

FWS Agreement Number: 70181BJ037
COOPERATIVE AGREEMENT

between

U.S. Fish and Wildlife Service, Region 7
1011 East Tudor Road
Anchorage, Alaska 99503-6199

and

Glen Liston
15621 SnowMan Road
Loveland, CO 80538

FINAL REPORT FOR THE PROJECT:

**Mapping Suitable Snow Habitat for Polar Bear Denning
Along the Beaufort Coast of Alaska**

INTRODUCTION:

Polar bears along Alaska's Beaufort Sea frequently give birth to young in land-based snow dens. These dens are established in November, typically in deep snowdrifts that have developed in the lee of cut-banks found along streams, rivers, and the coast. Durner et al. (2001, 2006) indicated that, for 24 known land den sites, the local slopes ranged from 15 to 50° and were 1.3 to 34 m high. The dens faced all directions but east. They published a distribution map based on habitat characteristics, presumably reflecting snow drifting, largely bracketing the generally northward flowing drainages of the region. No attempt was made in the cited studies to model snow drifting explicitly, though it was recognized that this was an important control on den distribution.

For the same region Benson (1982), Benson and Sturm (1993) and Sturm et al. (2001) have described snow drift growth and development. They note that in some years significant drift growth occurs before December, but in others this may not happen until February or March. They found drift location and geometry to be more consistent, however, with drifts of similar size and shape forming each year due to strong topographic controls, suggesting that the location and shape of drifts are generally invariant and that generic snowdrift maps could be established.

A number of analytical tools are available to simulate snowdrifts resulting from blowing snow processes (Liston and Sturm, 1998; Liston and Sturm, 2002; Liston and Elder, 2006a, b; Liston et al., 2007; Liston and Hiemstra, 2008). These tools, basically numerical models that mimic the physical interaction of snow and wind, determine when, how far, and how much snow will be transported and deposited in response to topographic variations. These may be the best tools available for simulating snow-drift accumulations and the associated potential bear denning sites.

The purpose of this project is to provide better information to industry and regulatory agencies regarding the likely locations of polar bear dens. This project integrates snow physics, high-resolution digital elevation data, and bear biology to produce more refined and accurate maps predicting suitable polar bear den habitat than are currently available. The work consists of data gathering, consultation between snow and bear scientists, modeling, and sensitivity studies to understand the various factors influencing den location and evolution along the Beaufort Coast.

The proposed work is intended to refine current methods of identifying polar bear denning sites by incorporating higher-resolution topographic data (obtained from LiDAR data collections) and applying an existing model of blowing and drifting snow deposition (SnowModel; Liston and Elder, 2006a) to more accurately predict the presence, timing, and evolution of snowdrifts suitable for bear dens. SnowModel is a numerical model that mimics the physical interaction of snow, wind, topography, and land cover, to determine when, how far, and how much snow is transported and deposited in response to topographic and land-cover variations. SnowModel has been well tested throughout the Arctic, and was originally developed for Arctic Alaska applications. Application of these tools will allow us to define the time-evolution and locations of potential bear denning sites, and the controlling factors that determine key polar bear denning characteristics.

The project is considered a “proof-of-concept” investigation to test whether significant improvement over current models is achievable. If the results are promising, an implementation phase will require further investment in a decision-support tool that ingests available topographic and weather data, and provides an output map of high-probability den locations.

PROJECT TASKS:

TASK 1: *Process digital elevation model (DEM) and land-cover data distributions in preparation for MicroMet and SnowModel simulations.*

To run MicroMet and SnowModel, topography and land-cover datasets were required on common grids (coincident spatial area and grid increment) over the domain of interest. The LiDAR topography data provided for this project were not spatially continuous (e.g., data were missing over coastal river deltas, lakes, and areas approximately 3-km inland of the coast). Therefore, to prepare the topography and land-cover datasets to be used in the snow-drift model simulations, the following steps were taken:

- 1) National Elevation Data on a 2 arc-second grid (NED2; approximately 50 m grid increment) for northern Alaska were obtained. The associated TIF files were converted to ASCII grid files using the gdal_translate program included within the Geospatial Data Abstraction Library. The longitude-latitude coordinates were converted to UTM coordinates (Zone 6) using the GRASS GIS program and the NED2 data was regridded to a 25-m grid using the Barnes Objective Analysis scheme.
- 2) The coastal Alaska LiDAR topography data were provided to InterWorks Consulting as a collection of 2-km by 2-km GeoTIFF files. These were converted to ASCII grid files using the gdal_translate program, and all of the files were merged into a single data file (Fig. 1).

- 3) The NED2 data were resampled to the LiDAR 2.5-m grid. A vertical offset of 5.48 m was added to all of the LiDAR data values to eliminate elevations below sea level. The two data files were then merged into a single file (LiDAR data replaced the NED2 data wherever LiDAR data were available).
- 4) To correct step-changes in elevation at the boundaries of the NED2 and LiDAR data (Fig. 2), an edge-smoothing algorithm was developed and implemented.
- 5) The domain was then clipped to allow a focus on the western half of the LiDAR domain, and to make the array sizes manageable (from 89600x28800 to 42000x21000 array elements) (Figs. 3 and 4).
- 6) To develop a land-cover dataset over this domain, National Land Cover Data (NLCD) on an Albers, 30-m projection was obtained for Arctic Alaska. The associated GeoTIF file was converted to an ASCII file using the gdal_translate program, and the coordinates were converted to UTM using the GRASS GIS program. This was regridded to a 25-m grid and then resampled to the 2.5-m LiDAR data grid (Fig. 5). The NLCD land cover classes were then converted to the classes required by MicroMet and SnowModel (Table 1).
- 7) The efforts summarized above produced two topography and land-cover datasets for the Figs. 4 and 5 domain: one at a 25-m grid increment (a 4200x2100 element array; 8.82 million grid cells), and one at a 2.5-m grid increment (a 42000x21000 element array; 882 million grid cells).

TASK 2: *Process local meteorological data in preparation for MicroMet and SnowModel simulations.*

To run MicroMet and SnowModel, meteorological datasets were required to force the models. The following steps were taken to produce the required atmospheric forcing for the model simulations:

- 1) Meteorological station data was obtained for the years 2008 through 2011 for the available stations within and near the Fig. 4 simulation domain (Table 2). In addition, to define winter precipitation inputs, Natural Resource Conservation Service (NRCS) SNOpack TELelemetry (SNOTEL) data and North American Regional Reanalysis (NARR) data for the Fig. 4 area were obtained.
- 2) Longitude – latitude coordinates of these stations and the NARR reanalysis were converted to UTM coordinates using the GRASS GIS program (Table 2). Using this information, in addition to the meteorological values (air temperature, relative humidity, wind speed, wind direction, and precipitation), the datasets were cast in the form required by MicroMet to perform the model simulations. A plot of the station locations, corresponding to the Fig. 4 simulation domain, is provided in Fig. 6, where the stations ID Numbers correspond to those listed in Table 2.
- 3) To define the precipitation inputs for the model simulations, and avoid the temporally missing-data issues associated with the SNOTEL observations, the following analysis was implemented. Defining the snowfall season to span September through April, SNOTEL data was compared with NARR precipitation outputs. For the 3 winters 2008-2009, 2009-2010, and 2010-2011, during September through April, SNOTEL precipitation (snowfall) averaged 114 mm per year, and NARR water-equivalent

precipitation (snowfall) averaged 68 mm per year. Therefore, in these simulations, NARR precipitation values were multiplied by 1.7 ($=114/68$). Therefore, NARR was used to define the timing of snowfall events, and was scaled to more realistically represent the observed magnitudes found in this region.

- 4) The following plots summarize the meteorology in this area. This meteorology is directly linked to the evolution of the snow drifts used by denning polar bears. Fig. 7 displays the air temperature for all of the stations (ID Numbers 1-9; the different colors correspond to the different stations) listed in Table 2, showing that all stations measure similar air temperatures. Figs. 8, 9, and 10 indicate that all stations measure similar relative humidities, wind speeds, and wind directions, respectively. In this relatively flat area of the Arctic, the meteorological fields are relatively uniform. The wind speed variations in Fig. 9 are largely the result of the towers and instruments being at different heights (ranging from 2 to 10 m); something that has not been corrected for in this figure (note that logarithmic height corrections would lead to an approximately 15% difference among the stations). Fig. 7 suggests air temperatures are low enough (below approximately 2 °C) to have snowfall starting in September and continuing throughout the winter. Fig. 8 shows that the humidity in this area is fairly high; this means that sublimation is minimal and any precipitation falling as snow is available to be transported by the wind (large quantities of snow do not sublime back into the atmosphere). Fig. 9 indicates that the winds in this area are frequently sufficient to transport snow (the wind-transport threshold for blowing snow is approximately 5 m/s in this area). Fig. 10 shows that there are two dominant wind directions in this area of the Arctic coast: winds approximately from the NE to E (45.0 to 90.0 degrees) and from the WSW to SW (247.5 to 270.0 degrees). These wind directions impose a critical control on the formation and location of the bear-denning snow drifts. As a further refinement of the wind direction analysis, Fig. 11 displays the wind directions for winds over 5 m/s. Fig. 12 plots the wind directions for winds over 10 m/s, and Fig. 13 shows the wind directions for winds over 15 m/s. These figures highlight how the higher wind speeds associated with storm events come predominantly from the NE to E and WSW to SW directions. These higher wind speeds transport the most snow (there is non-linear increase in wind transport with increasing wind speed, above the approximately 5 m/s threshold), assuming snow is available to be transported. Figs. 14, 15, and 16 compare the meteorological observations from Deadhorse Airport and NARR for air temperature, wind speed, and wind direction, respectively. For air temperature, the NARR data typically has a cold bias during summer (presumably tied to its cloud-cover representation), but is judged to be representative enough for winter snow simulations. The wind speeds are underestimated by NARR. Because of the sensitivity of snow redistribution to wind speed, some kind of bias correction would be required to use NARR data in this area to force the blowing snow simulations. The NARR data appears to be a reasonable representation of wind direction, and is likely an appropriate dataset for historical wind direction analyses. Fig. 17 compares Deadhorse Airport and NARR wind direction data for wind speeds over 5 m/s. The relatively low NARR wind speeds are evident in the plotted data densities (there are fewer NARR data points plotted above the 5 m/s threshold), suggesting that a wind-speed bias correction is needed before any wind-direction analysis is performed. Fig. 18 displays a comparison of SNOTEL and NARR precipitation data. Both datasets suffer from different weaknesses, and neither one is considered a very accurate representation

of the natural system; for example, SNOTEL has resolution and delayed-timing issues (the totalizing instrument only measures in increment sums of 0.1 inch/day, or 0.106 mm/hr), and NARR typically ‘drizzles’ a little at every time step (there are rarely hours of zero precipitation in its data record). These are common and well-known problems with these datasets. In spite of these limitations, the SNOTEL and NARR precipitation data appear to be sufficient to provide snow-precipitation inputs for the model simulations performed as part of this project. Of particular note (from the combination of Figs. 7 and 18) is the fact that solid precipitation (snow) falls during the months of September, October, and November. This, in conjunction with the previous figures, suggests there is both snow and sufficient wind to transport snow. This combination is required if snow drifts are to grow to sufficient depth by late November, the approximate date by which mother polar bears dig and occupy their dens.

- 5) The above meteorological datasets were merged into the file format required for MicroMet and SnowModel simulations. Both hourly and daily files were created for the winters 2008-2009, 2009-2010, and 2010-2011.

TASK 3: *Configure MicroMet and SnowModel to perform the required simulations.*

The MicroMet and SnowModel setup and parameter file (snowmodel.par; Table 3) provides an example of the model configuration used for the model simulations. To reduce the computational expense and memory requirements to manageable levels, the model was run on a daily time step, for the period 1 September 2010 through 1 July 2011, over two subdomains defined within the Fig. 4 domain. The simulation domains were 9000 by 6000 grid cells in size, or 54 million grid cells (Fig. 19). Further analysis of the final topography data for these two domains indicated that additional topographic offsets were required in order to define an appropriate sea level (the vertical datum was not defined or corrected in the original LiDAR datasets provided to the project). To correct this, all land points were lowered by 1.6 m and 2.6 m (determined by graphical/visual observation of the topography data near the coastlines), for the NW and SE simulation domains, respectively. This correction was required, or SnowModel would produce physically unrealistic drift features where there is a ‘topographic’ step-change defined in the ocean, *ocean-ward* of where the ocean meets the land, at the boundary of the LiDAR data and the ocean that was defined by the land-cover dataset. Note that these two simulation domains capture the majority of the observed historical terrestrial denning sites where LiDAR data exist.

TASK 4: *Perform high-resolution (2.5-m grid increment) MicroMet and SnowModel simulations of snow-drift related potential polar bear denning sites along Alaska’s Arctic coast.*

Each of the simulations took 30.6 hours of CPU time. The simulations output 19 meteorological- and snow-related variables during each day of the simulation (304 days). Each output variable array was approximately 65.7 GB in size for a total archive (for both simulations) of approximately 2.5 TB of data. One of the output fields simulated by SnowModel is the equilibrium snow-drift depths for each of the 8 primary wind directions (i.e., N, NE, E, SE, etc.). These represent the maximum possible snow depths that can form over the landscape in response to the given topographic variations, under conditions of sufficient snowfall and wind. Because there is much to be learned by looking at those fields, the following presentation analyzes those outputs in light of the meteorological analysis performed in Task 2.

A particular challenge to representing the model-simulated snow drifts graphically over the two rectangular domains identified in Fig. 19 occurs because the snow drifts are relatively small and infrequent features within a vast landscape. Fig. 20 displays the equilibrium (deepest possible) snow drifts, for the NW simulation domain, from both NE and SW winds. They are nearly too narrow to see. Fig. 21 presents the same data as Fig. 20 but for NE winds only, for the case where each grid cell containing a drift over 1 m deep has been expanded horizontally to 10 grid cells in all directions. This effectively adds a 25-m boundary around all model simulated drift traps in order to make them more visible in the figure. Fig. 22 highlights the drifts over 1-m deep, resulting from SW winds.

Figs. 23, 24, and 25 present the same information as that presented in Figs. 20, 21, and 22, respectively, but for the SW simulation domain.

In an effort to better visualize the snow distributions simulated by the model, Fig. 26 provides a zoomed-in picture of the data presented in Fig. 20. This figure displays the equilibrium snow drifts from both NE and SW winds, along with the historical denning site record. As anticipated, den locations must correspond to sufficiently deep snow drifts, and any mismatch in these two variables must indicate errors in one or both of them. Three likely possibilities present themselves: errors in the den location GPS measurements (the denning dataset spans from 1910 to 2011); offset errors from where dens were located with some remote-sensing device (e.g., hand-held FLIR when traveling on the sea ice along the coast); and LiDAR-retrieval geo-location errors.

Fig. 27 provides a cross-sectional profile of the line given in Fig. 26, and shows snow drifts for both NE and SW winds. In order to form a 1.5-m deep drift in a cut-bank that is only 3-m high, the cut-bank must be quite steep. In this location, it is steep enough in the lee when there are NE winds (a 1.5-m drift), but not in the lee of SW winds (a 0.75-m drift).

Fig. 28 displays a zoom-in of the drifts presented in Fig. 26. The correspondence between the drift location and denning locations is remarkable. Note the clear GPS errors in den-site location; it is just not possible for dens to be located in the flat, snow-erosion areas on top of the island (it is possible that there is some real terrain feature that creates a drift in the natural system that was not captured by the LiDAR acquisition). Also note that while these bear dens are located in the simulated drift locations, these are not very deep drifts; they are 1.5-m deep, or less. Fig. 29 provides a visual image of an Arctic Alaska coastal snowdrift that is similar to that simulated by SnowModel in Fig. 28. This image was collected a few weeks into spring snowmelt, after the thin veneer snow (typically 10 to 40 cm deep at end of winter) had melted and all that remained were the deeper snowdrifts.

Fig. 30 presents the cross-sectional profile of the line given in Fig. 28. Shown are the topography, the snow lying over the topography, and the snow-depth profile. Note that if the 3.0-m cut-bank was vertical, then the drift would be 3.0-m high, but because the cut-bank has some significant slope, the actual snow depth in the drift is reduced to a maximum of approximately 1.5 m. Fig. 31 provides another perspective on the drift in Fig. 30. In addition to zooming-in on the drift itself, note that the distance along the x axis is 140 m, and the y axis is 4 m. To provide

an image of the 1-to-1 aspect ratio found in nature, either the y axis would have to be reduced to 1/4th the current height, or the x axis would have to be 4 times longer; in the natural system these drifts are very long and thin. This is part of the ‘equilibrium drift’ definition, where the snow has filled the lee of the cut-bank to the point where the snow surface is relatively level coming down the lee side of the top lip of the cut-bank, and it continues dropping at only a slight angle until it reaches the distant slope downwind. When the wind and snow blows over this shallow, negative-angled (relative to the wind direction) slope, the wind speed is not reduced and any snow carried by that wind continues past the (now full) cut-bank (drift trap). In other words, the drift trap is full. Fig. 32 displays pictures collected along Alaska’s Arctic coast in the summer and winter, showing a typical coastal cut-bank in summer, and a cut-bank full of snow during winter.

Fig. 33 presents periodic excerpts from a movie made using outputs from an idealized SnowModel snowdrift simulation. The simulation assumes a snowstorm on the 15th of every month; the mother polar bear occupied the den on 1 December; she had two cubs on 1 January; the cubs grew during the next few months; and they all emerged from the den in early April. The full movie is available in the education and outreach lesson and presentation discussed in Task 5.

TASK 5: *Develop and implement an education and outreach program that includes a Project-Based Learning presentation on polar bear denning issues.*

Ms. April Cheuvront, an 8th grade science teacher in North Carolina, developed an interactive Keynote (Mac) and PowerPoint (PC) presentation on polar bears, their denning practices, and how they are changing over time. The presentation includes pictures of polar bears, maps of observed denning sites, and pictures and movies made from model simulations of time-evolving snowdrift evolution and potential denning locations developed as part of this project. In addition, she developed a paper-lesson that teachers and students can use to map polar bear denning sites using longitude and latitude coordinates along Alaska’s Arctic coast.

To acquire the expertise required to develop these lessons, Ms. Cheuvront attended the 20th International Conference on Bear Research in Ottawa, Canada, July 2011, where she spent hours talking to several internationally recognized bear experts at the conference, such as Ian Stirling, Steven Amstrup, and Dick Shideler. Also at this conference, Ms. Cheuvront established a connection with Polar Bears International (PBI) that led to a memorandum of understanding which allowed her to use PBI’s professional-quality photographs and videos in her lessons and presentations. Furthermore, numerous reports and refereed journal papers were read and studied to gather the material required for the lessons. In addition, during the fieldwork activities (February and April 2012) associated with this FWS project’s sister project [funded by the National Fish and Wildlife Foundation (NFWF)], Ms. Cheuvront collected supplementary information on polar bears, dens, and snowdrifts through her interactions with Craig Perham, FWS, and Dick Shideler, Alaska Department of Fish and Game.

Ms. Cheuvront presented her lesson to her 8th grade science classes during two school years (~230 students), and she taught ~250 kindergarten-through-12th grade students in Fairbanks, Alaska about polar bears. During the NFWF fieldwork, PBI hosted Ms. Cheuvront’s weblog on their website under their “Scientists and Explorers Blog” webpage. Her PBI weblogs received 1400 hits from her North Carolina school website alone. In addition, Ms. Cheuvront presented

the lessons at the 2012 National Science Teachers Association conference in Indianapolis, Indiana; thus making it available to other teachers nationwide (~25,000 participants). As part of NFWF funding, during October 2012 these lessons will again be presented in Fairbanks, Alaska schools, and in schools in the Alaska villages of Barrow, Wainwright, Nuiqsut, and Kaktovik. These lessons were printed on flash cards (100 copies made; total size of material 2.8 GB; Fig. 34) and are available for distribution to interested education and outreach personnel.

PROJECT PERFORMANCE AND CONCLUSIONS:

The ultimate deliverables for this project are proof-of-concept maps of potential polar bear denning sites along the Beaufort Sea Coast. These are provided as part of the discussions above and the figures associated with this report. Electronic versions of the datasets associated with these figures are also available by contacting Glen Liston. The general conclusion is that the approach presented herein, i.e., using local meteorological data and high-resolution (2.5-m horizontal grid increment) topography data, to drive a physically based snow-transport model, is valid and completely appropriate for this application. Additional observations garnered as part of this study include:

- Errors and limitations exist within the LiDAR dataset that limit its direct application to projects such as this. These include: 1) gaps in the data spatial distribution such as locations the LiDAR was not flown (e.g., river deltas) and water bodies where the LiDAR returned no data (e.g., lake and ocean areas); 2) lack of a vertical datum or definition of sea-level (there is no realistic 'zero' in this dataset); and 3) data restricted to a narrow, jagged corridor along the coast (Figs. 1, 2, and 3). In addition, because the snow distribution simulations are quite sensitive to abrupt topographic changes, any such errors in the topographic data will present themselves as anomalous drift features. For example, the long, linear drifts displayed in the SW area of Figs. 20, 21, and 22 clearly do not represent the natural system.
- In addition to adequate meteorology (e.g., sufficient wind and snow), topographic variability is required to create snow drifts of sufficient depth for polar bear dens. The provided LiDAR data resolution is sufficient to provide this topographic variability.
- There are sufficient meteorological data (air temperature, wind speed, wind direction, and precipitation) available in this area to analyze how these forcing fields have changed over the last 30+ years, and to drive MicroMet and SnowModel simulations of snowdrift formation. These are available from both weather stations and atmospheric reanalyses such as NARR.
- Winds that create snowdrifts along Alaska's Beaufort Sea Coast come predominantly from the NE to E and the SW to WSW.
- The snowdrifts are formed from the snowfall that occurs during September, October, and November (because the bears are typically in their dens by late November).
- The wind speeds and directions that occur during September, October, and November define which topographic slopes (e.g., SW or NE) the snow accumulates on, and are also expected to define which slopes the bears den on in any given year (the data are available to perform an analysis to see whether this is true).
- In this area, the coastal cut-banks that polar bears are denning near are barely high enough to create snowdrifts of sufficient depth to create dens. The cut-banks are often

- only 2-m to 3-m high, and this leads to snowdrifts that are only 1.5-m deep.
- Assuming that snow drifts have accumulated to equilibrium depths in response to NE and SW winds occurring during September, October, and November, the total area within the simulation domains that is suitable for denning habitat (defined to be snow depths over 1.25-m) is small: 0.047% (0.158 km²) of the NW simulation domain's 337.5 km² area and 0.021% (0.070 km²) of the SE simulation domain's 337.5 km² area (these numbers could be roughly doubled if you only consider land areas instead of the entire simulation domains). In addition, the actual denning area on any given year could be even less than these values if the wind redistribution of snow from either or both directions does not build the snow drifts to the equilibrium depth.
 - There appears to be a direct correspondence between the snowdrifts simulated by SnowModel and the historical den-site locations (e.g., [Figs. 20, 21, 22, 23, 24, 25, 26](#), and [28](#)).

REFERENCES CITED:

- Benson, C. S., 1982: *Reassessment of Winter Precipitation on Alaska's Arctic Slope and Measurements on the Flux of Wind Blown Snow*, Geophysical Institute, University of Alaska Fairbanks, 1982.
- Benson, C. S., and M. Sturm, 1993: Structure and wind transport of seasonal snow on the Arctic slope of Alaska, *Annals of Glaciology*, **18**, 261-267.
- Durner, G. M., S. C. Amstrup, and K. J. Ambrosius, 2001: Remote identification of polar bear maternal den habitat in northern Alaska. *Arctic*, **54**, 115–121.
- Durner, G. M., S. C. Amstrup, and K. J. Ambrosius, 2006: Polar bear maternal den habitat in the Arctic National Wildlife Refuge, Alaska. *Arctic*, **59**, 31-36.
- Liston, G. E., and K. Elder, 2006a: A distributed snow-evolution modeling system (SnowModel). *J. Hydrometeorology*, **7**, 1259-1276.
- Liston, G. E., and K. Elder, 2006b: A meteorological distribution system for high-resolution terrestrial modeling (MicroMet). *J. Hydrometeorology*, **7**, 217-234.
- Liston, G. E., and C. A. Hiemstra, 2008: A simple data assimilation system for complex snow distributions (SnowAssim). *J. Hydrometeorology*, **9**, 989-1004.
- Liston, G. E., and M. Sturm, 1998: A snow-transport model for complex terrain. *J. Glaciology*, **44**, 498-516.
- Liston, G. E., and M. Sturm, 2002: Winter precipitation patterns in arctic Alaska determined from a blowing-snow model and snow-depth observations. *J. Hydrometeor.*, **3**, 646-659.
- Liston, G. E., R. B. Haehnel, M. Sturm, C. A. Hiemstra, S. Berezovskaya, and R. D. Tabler, 2007: Simulating complex snow distributions in windy environments using SnowTran-3D. *Journal of Glaciology*, **53**, 241-256.
- Sturm, M., G. E. Liston, C. S. Benson, and J. Holmgren, 2001: Characteristics and growth of a snowdrift in arctic Alaska, U.S.A., *Arctic, Antarctic and Alpine Research*, **33**, 319-329.

FIGURES:

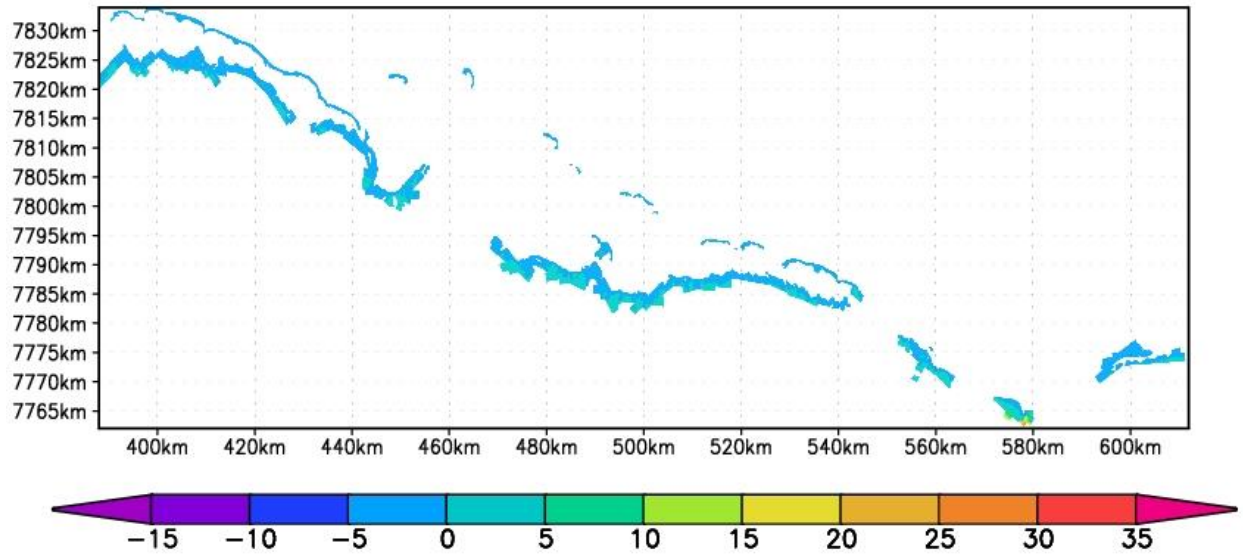


Fig. 1: LiDAR DEM (m) provided by FWS for this project.

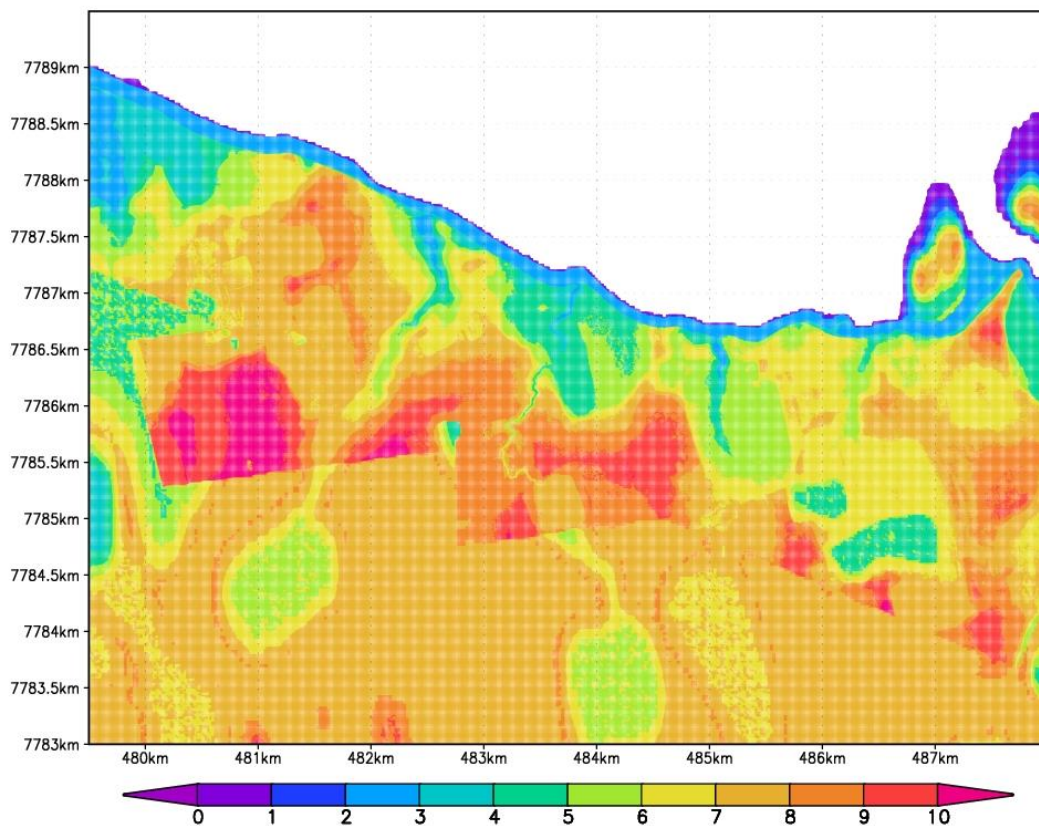


Fig. 2: Sample merged LiDAR (2.5-m grid increment) and NED2 (~50-m grid increment) topography data (m). Note the well-defined boundary between the different datasets. An edge-smoothing algorithm was implemented to reduce this sharp demarcation and prevent SnowModel from producing anomalous and unrealistic snow drifts along this edge.

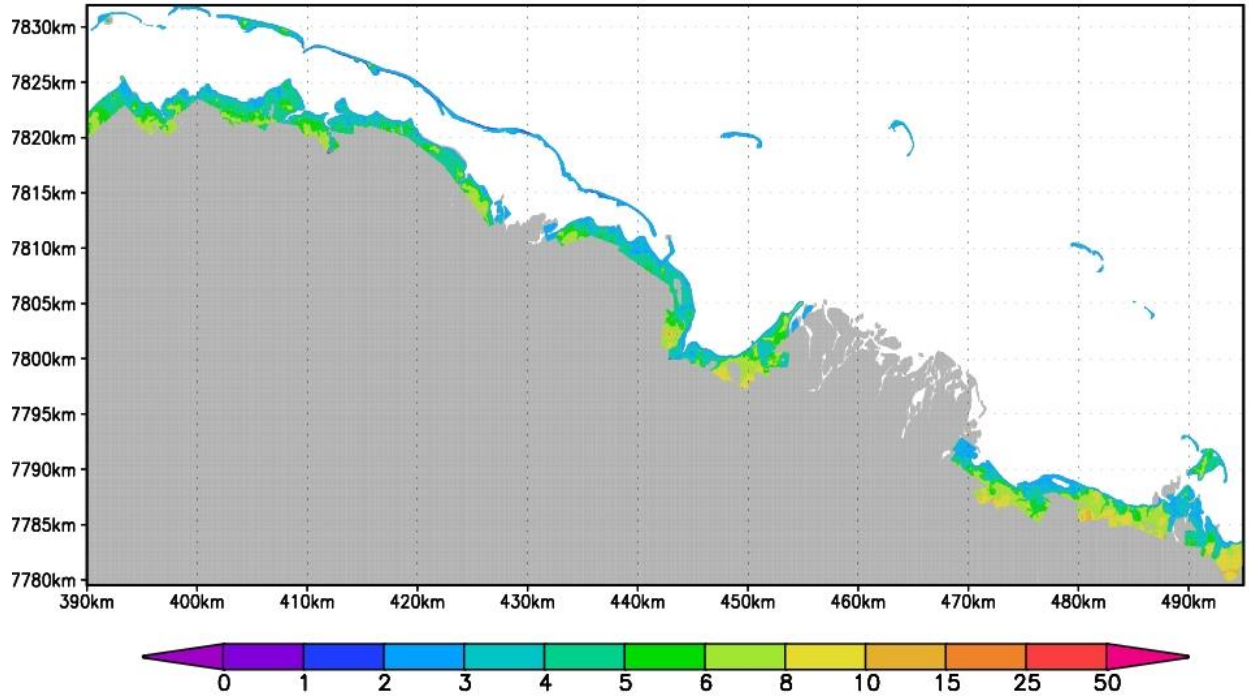


Fig. 3: LiDAR (2.5-m grid increment; color shades) and NED2 (~50-m grid increment; grey) DEM (m) sub-domains.

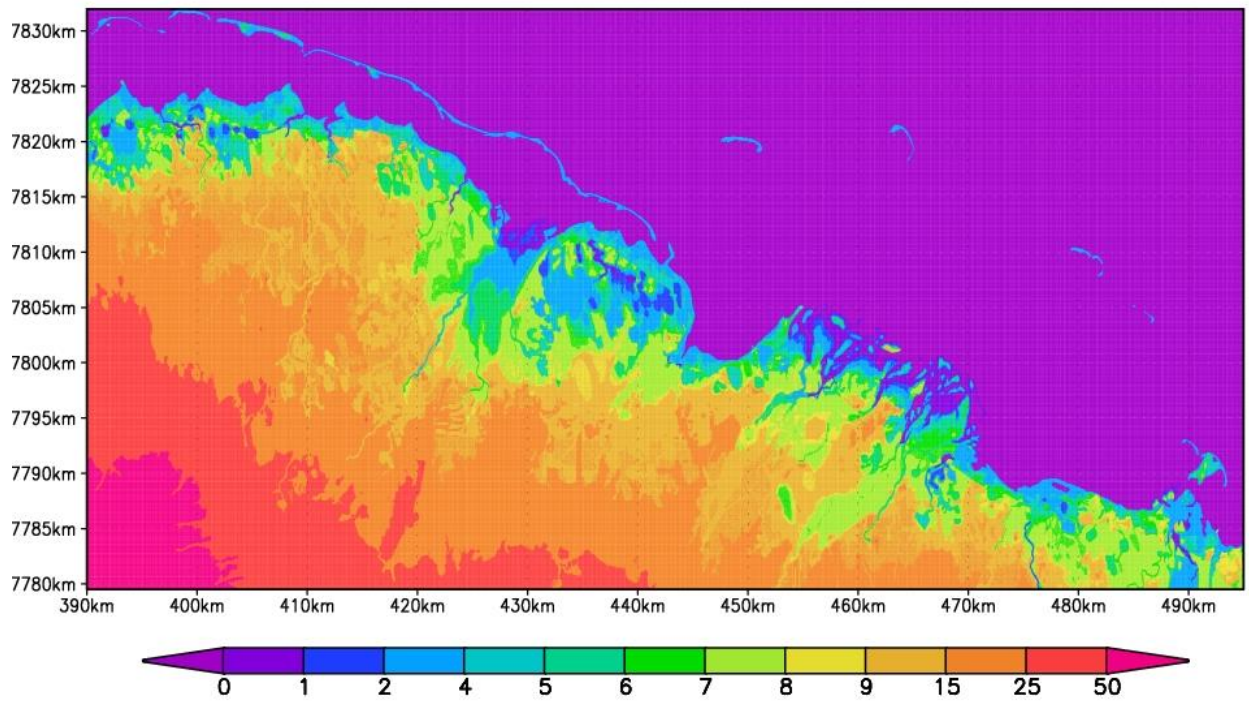


Fig. 4: Final, merged LiDAR and NED2 DEM (m) distribution (2.5-m grid increment).

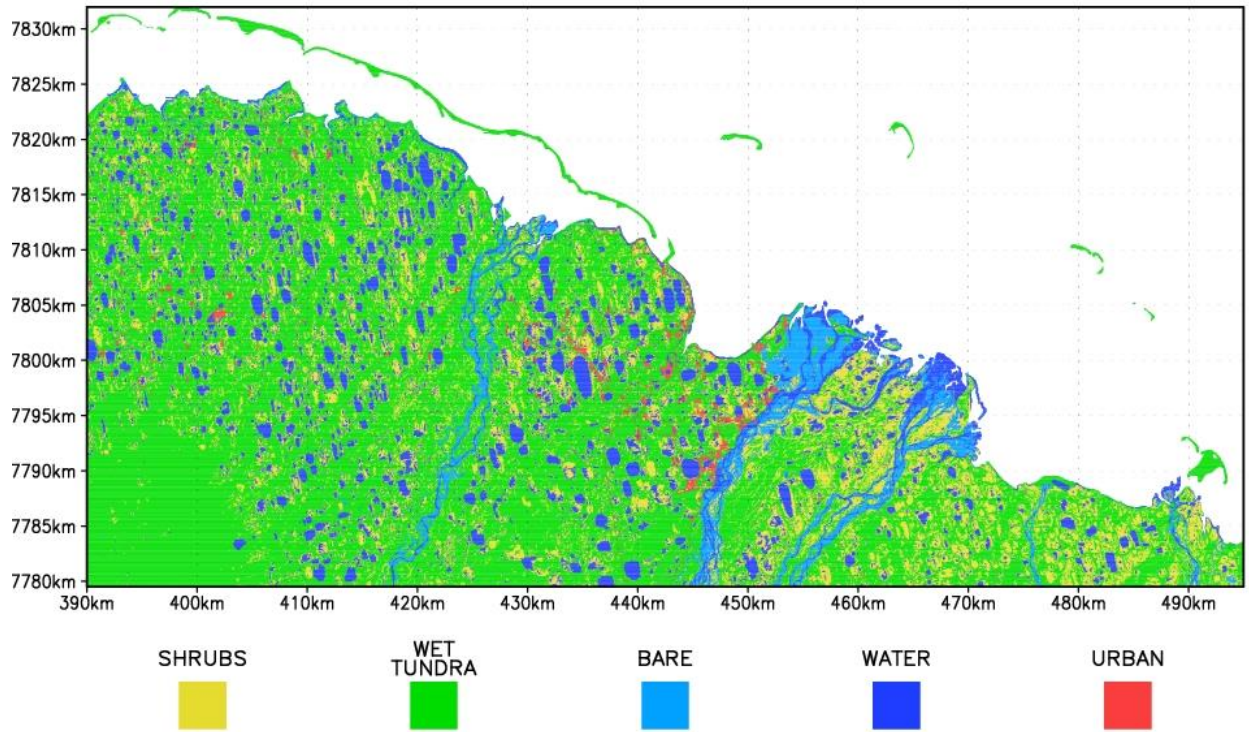


Fig. 5: NLCD land-cover distribution with SnowModel land-cover classes (2.5-m grid increment; 30-m resolution/data input).

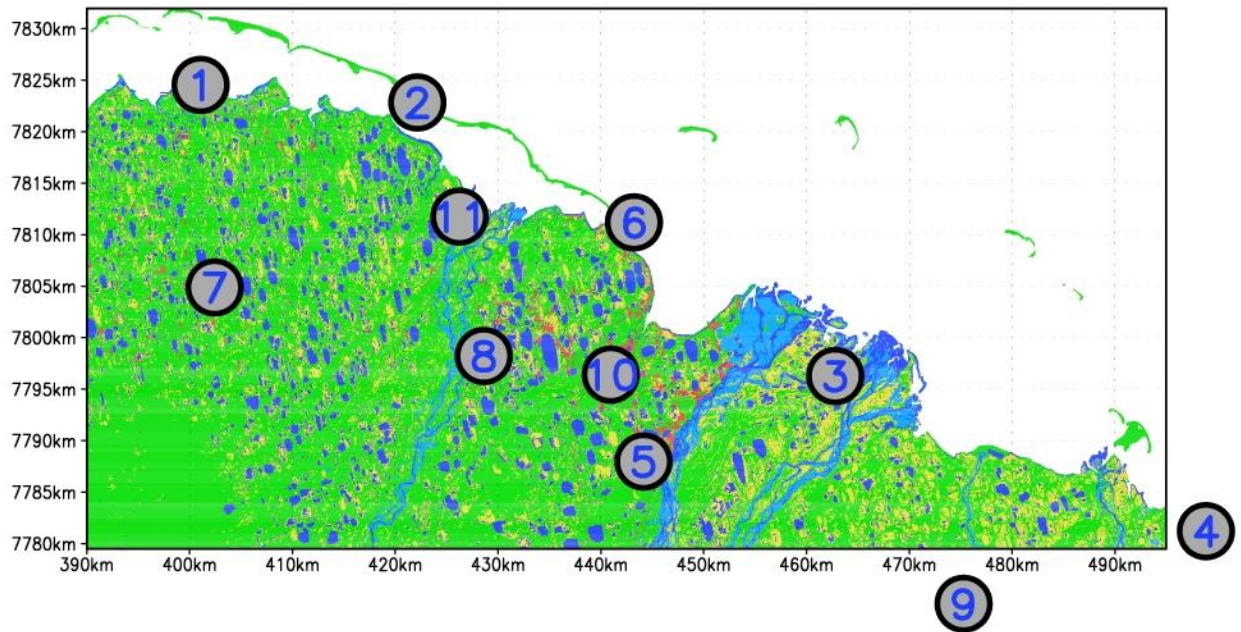


Fig. 6: Locations of meteorological stations located within the simulation domain. ID Numbers correspond to those given in Table 2.

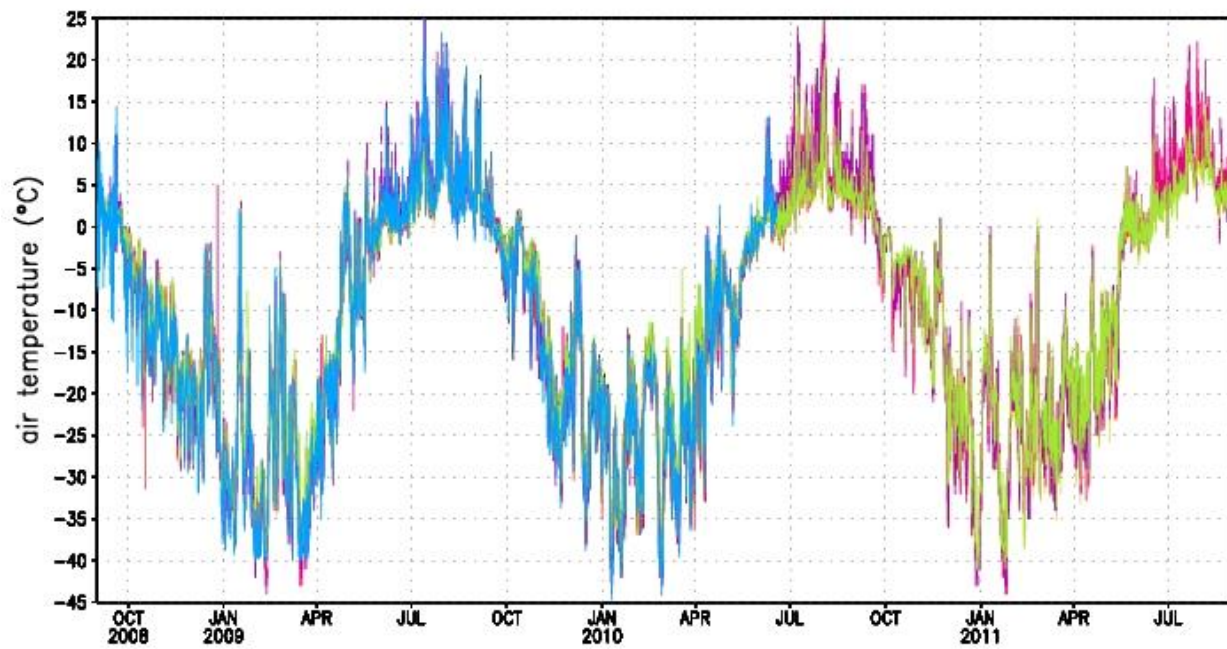


Fig. 7: Air temperature from all meteorological stations listed in Table 2.

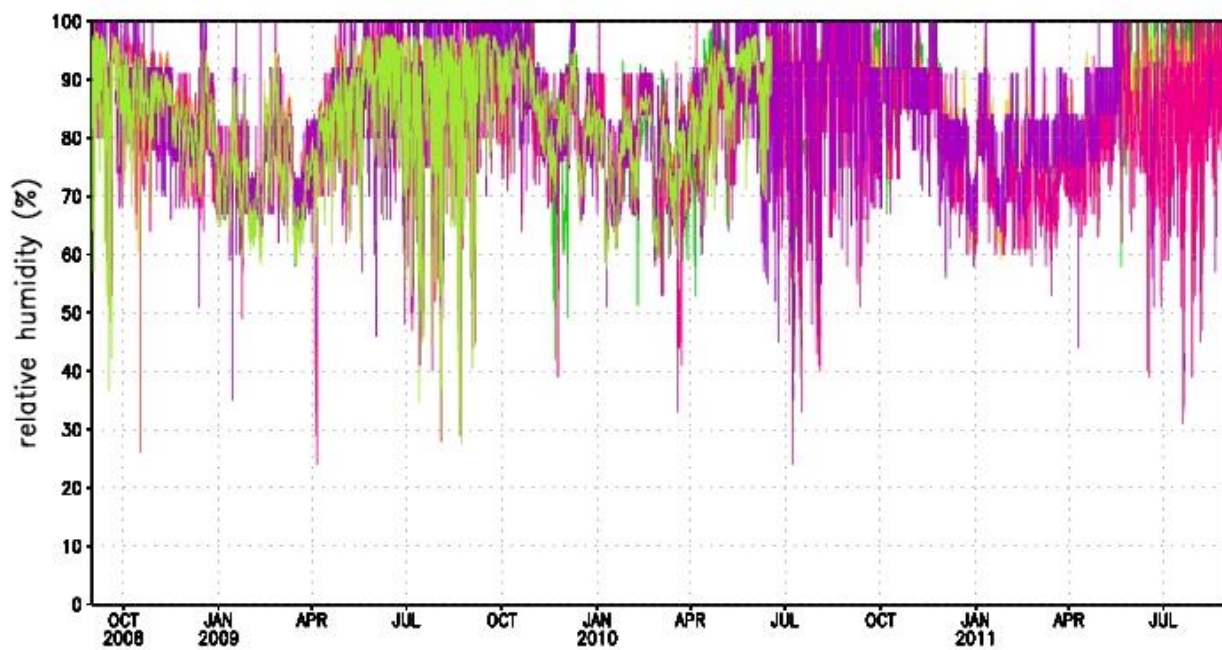


Fig. 8: Relative humidity from all meteorological stations listed in Table 2.

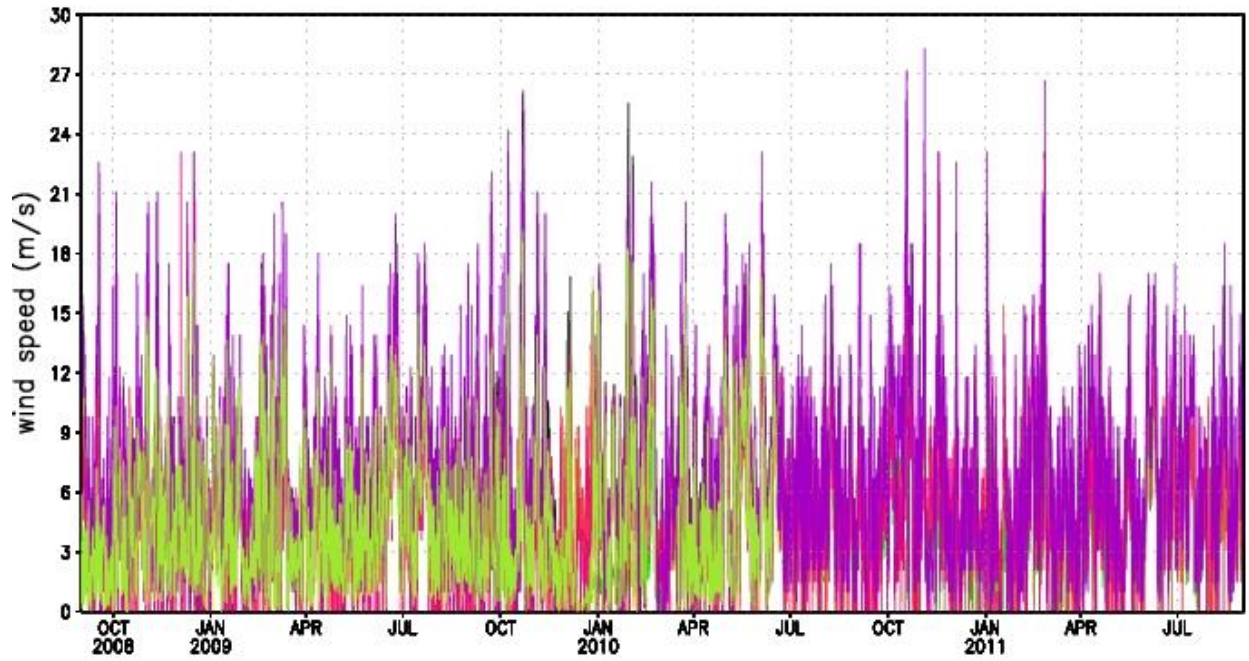


Fig. 9: Wind speed from all meteorological stations listed in Table 2.

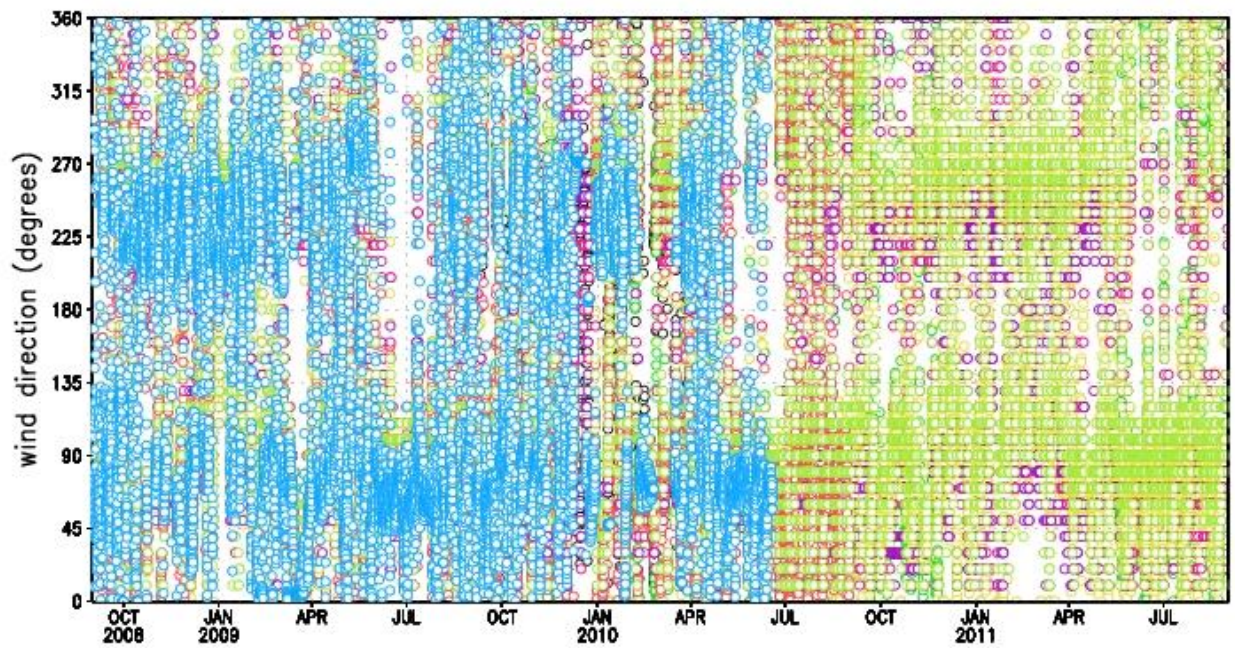


Fig. 10: Wind direction from all meteorological stations listed in Table 2.

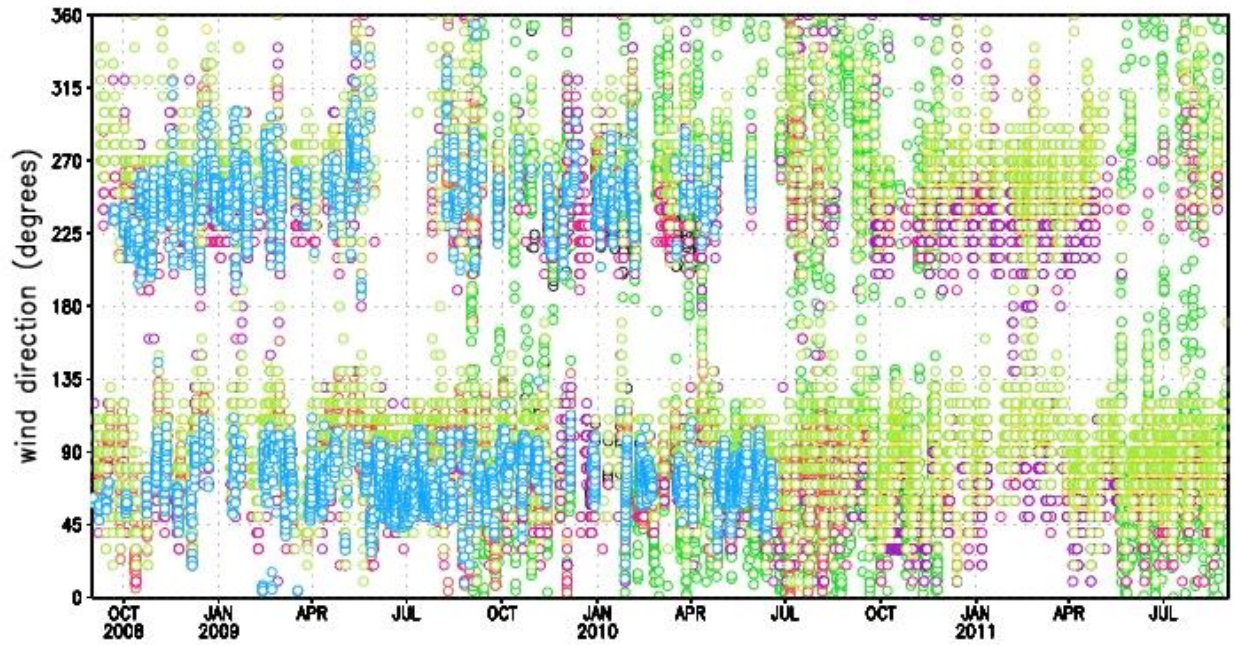


Fig. 11: Wind direction, for wind speeds greater than 5 m/s, from all meteorological stations listed in Table 2.

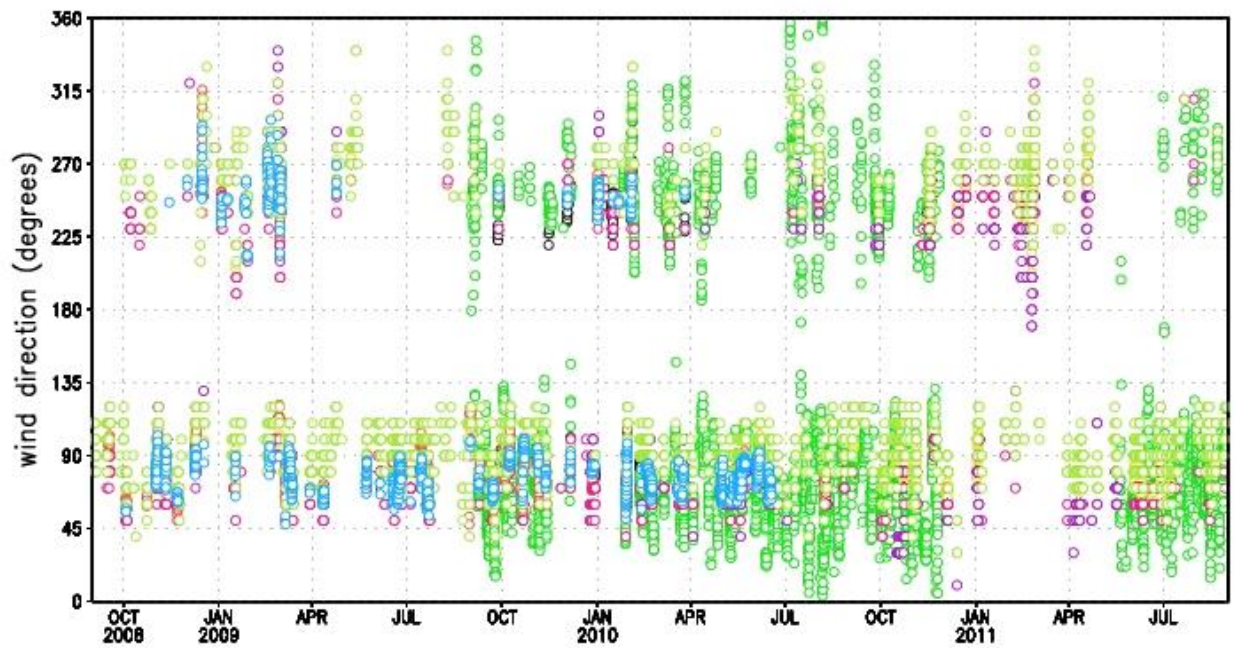


Fig. 12: Wind direction, for wind speeds greater than 10 m/s, from all meteorological stations listed in Table 2.

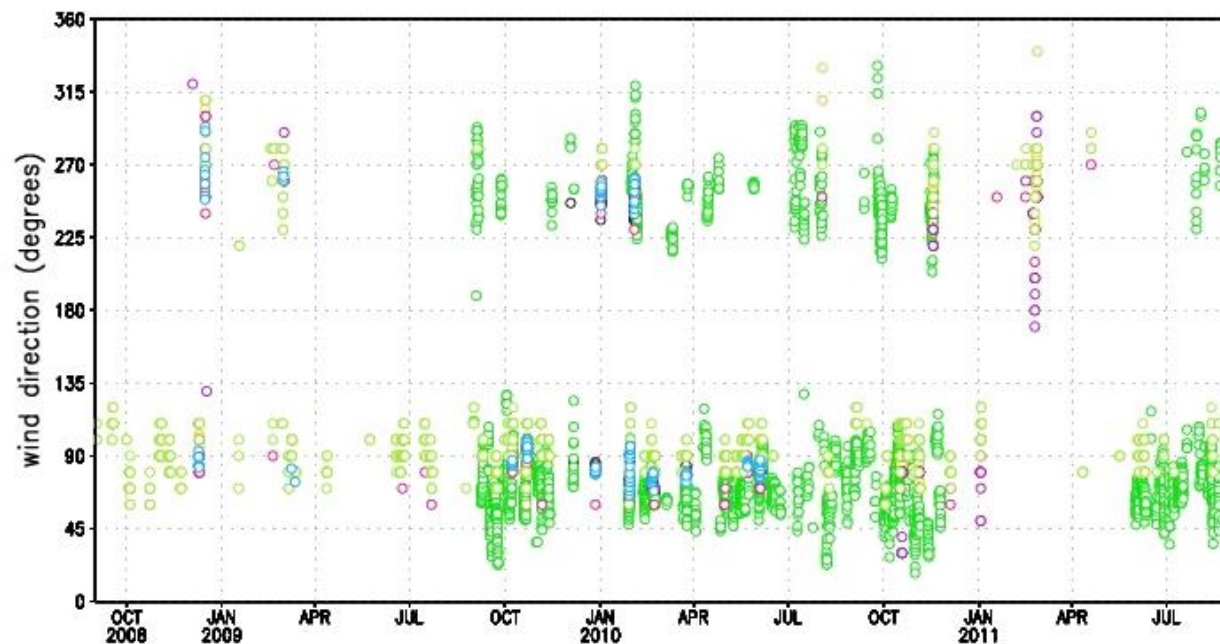


Fig. 13: Wind direction, for wind speeds greater than 15 m/s, from all meteorological stations listed in Table 2.

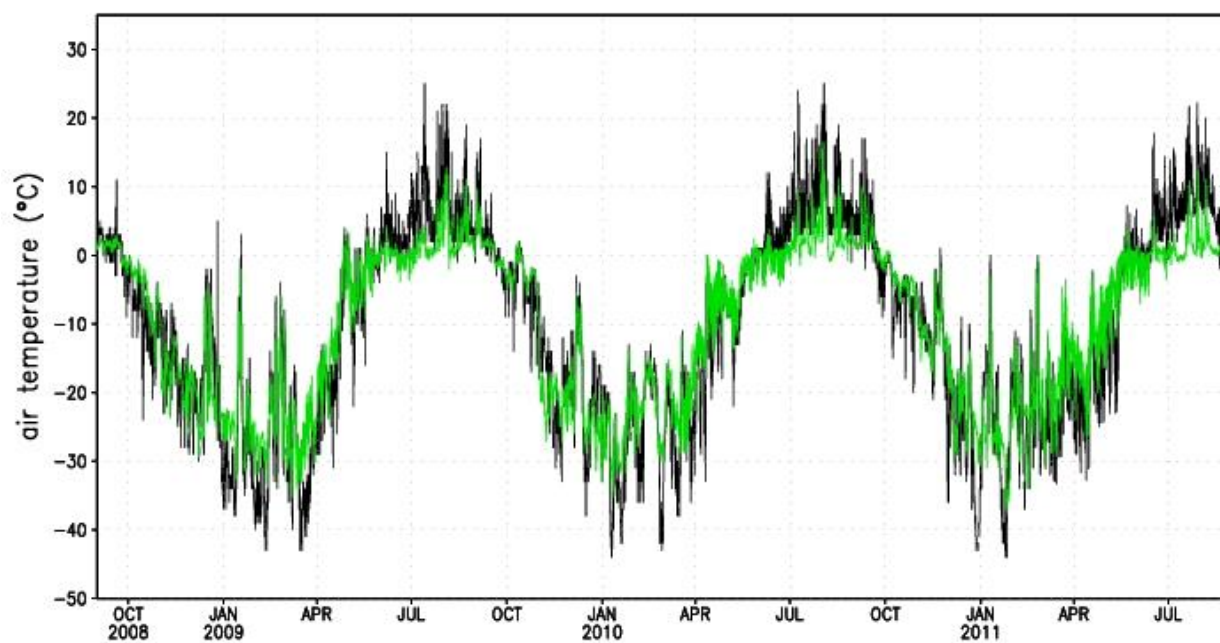


Fig. 14: Air temperature from Deadhorse Airport (black) and NARR (green).

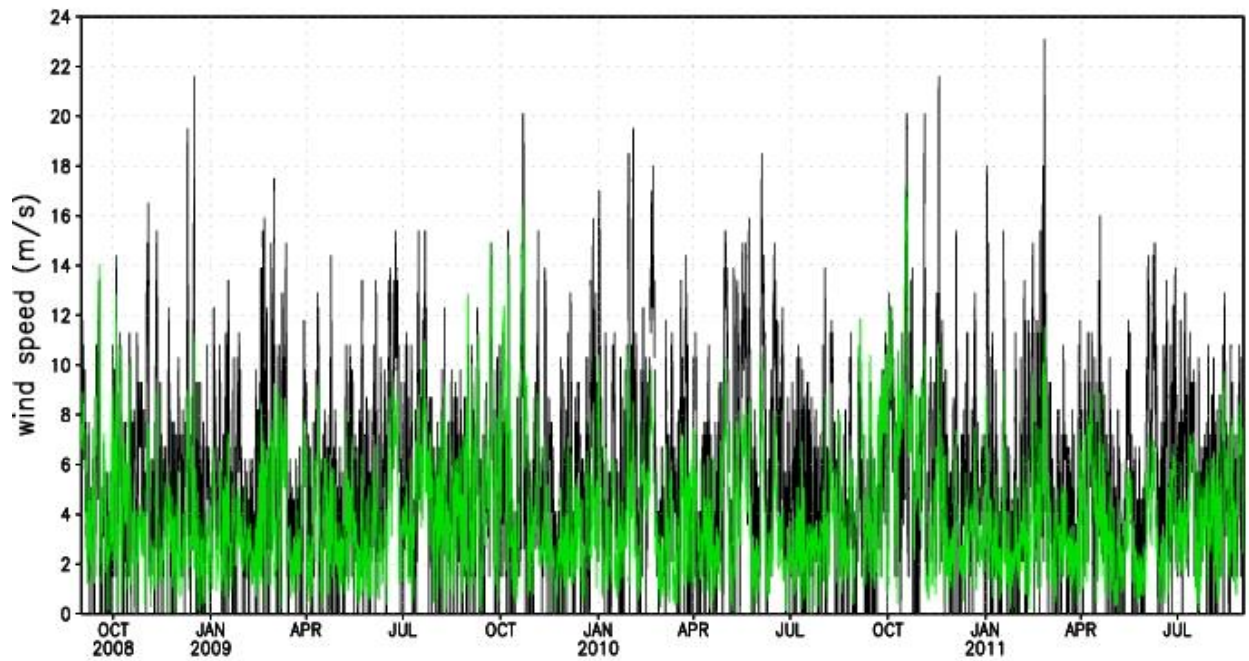


Fig. 15: Wind speed from Deadhorse Airport (black) and NARR (green).

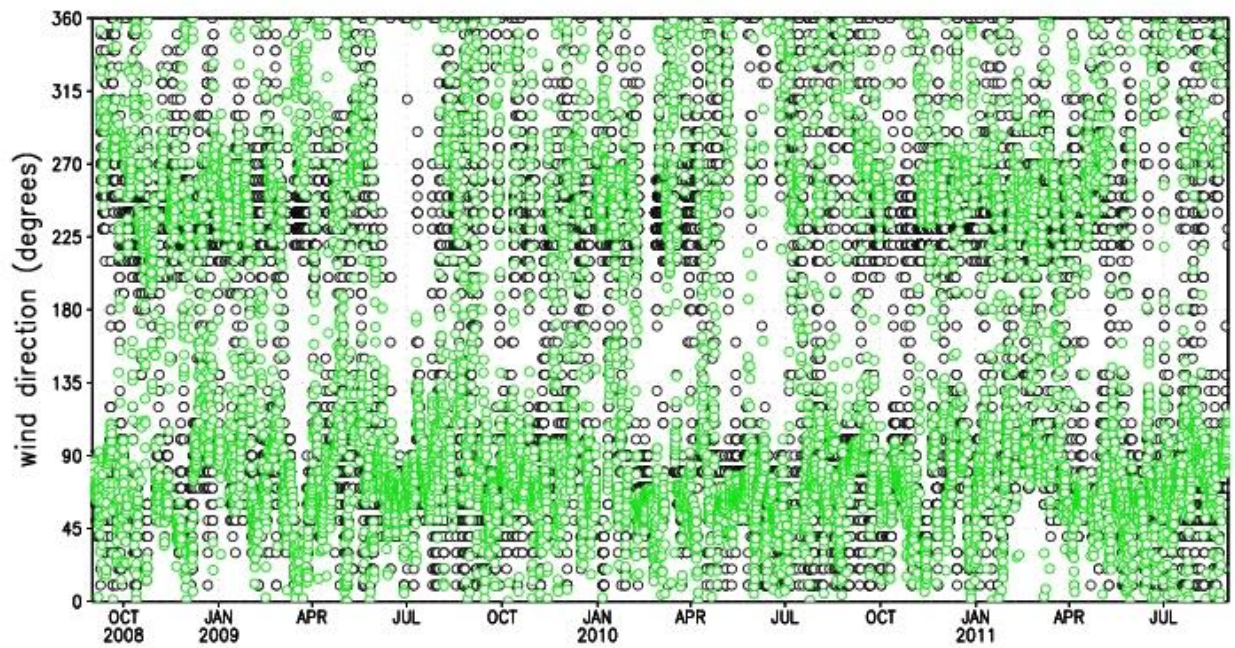


Fig. 16: Wind direction from Deadhorse Airport (black) and NARR (green).

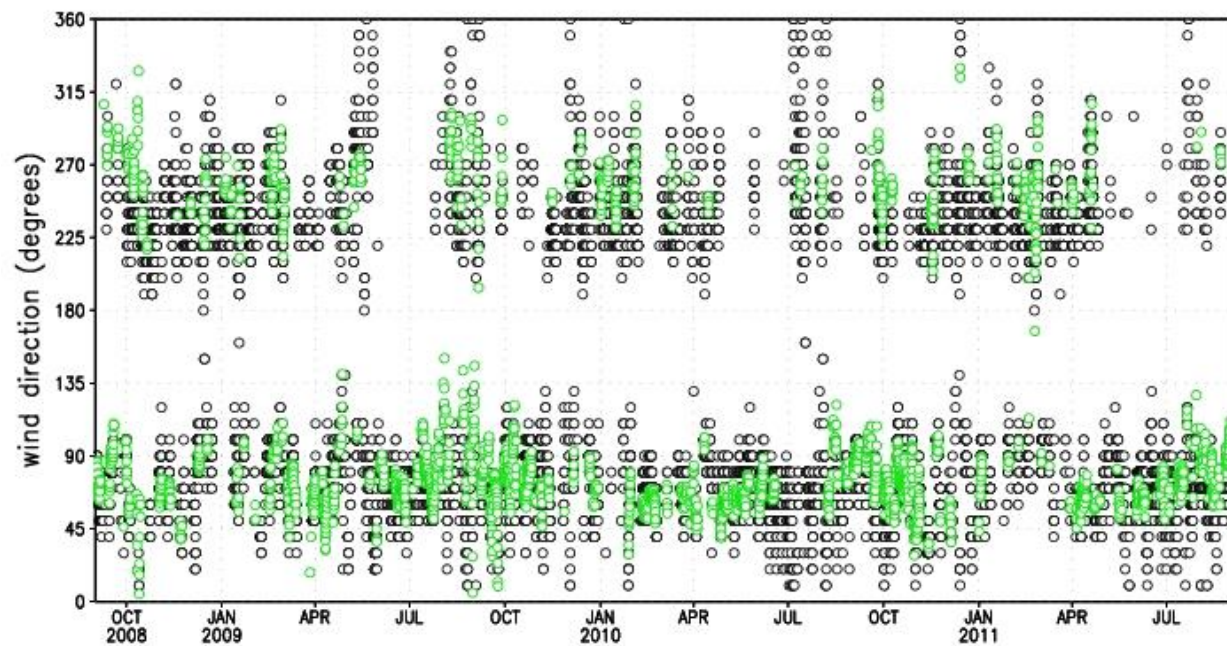


Fig. 17: Wind direction, for wind speeds greater than 5 m/s, from Deadhorse Airport (black) and NARR (green).

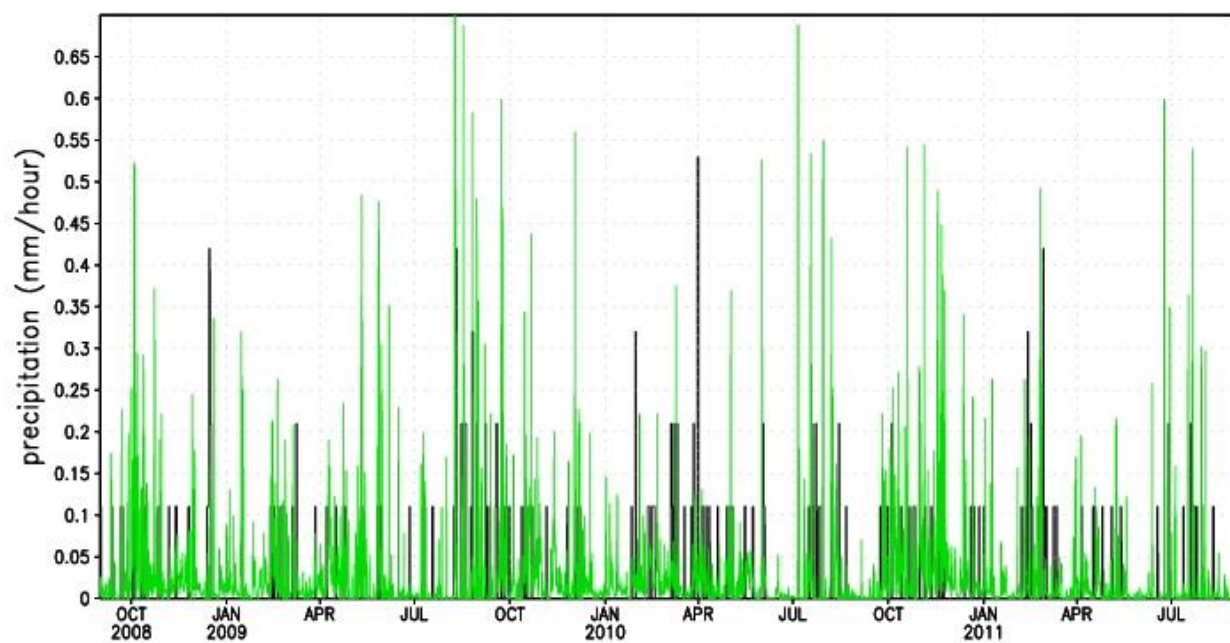


Fig. 18: Precipitation from the SNOTEL site (black) and NARR (green).

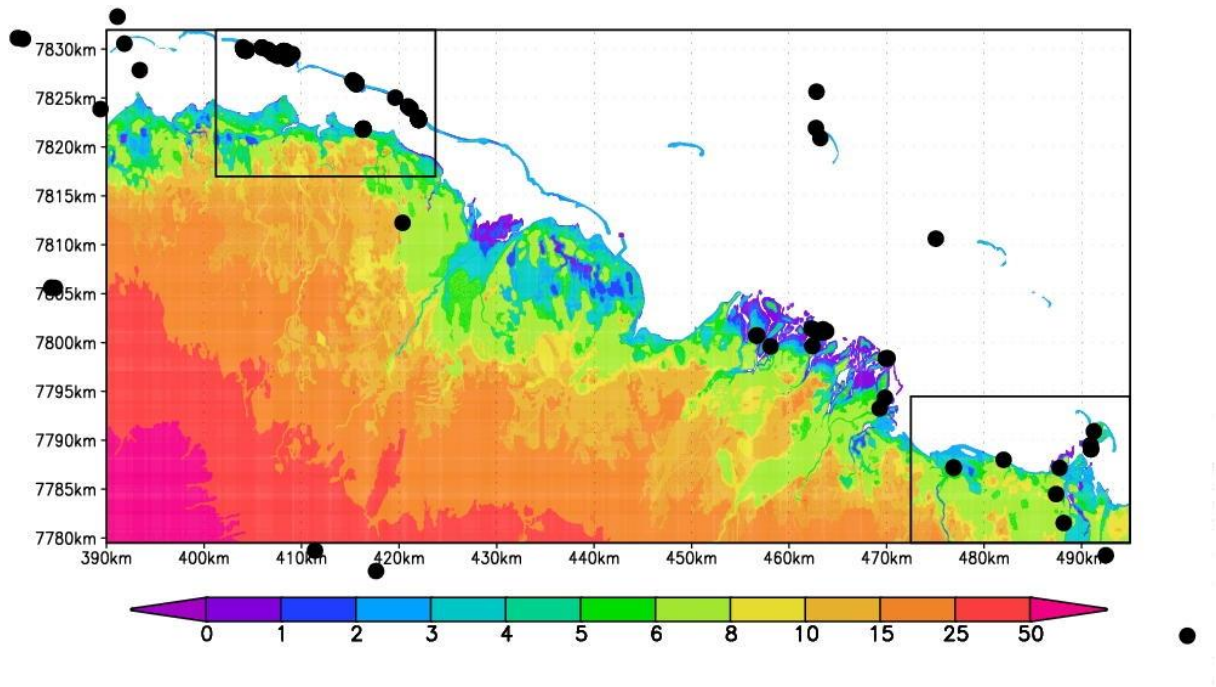


Fig. 19: Simulation focus areas (rectangular boxes), and historical polar bear denning sites (black dots; source, Craig Perham and USGS). Color shades are topography (m). Note that high-resolution LiDAR data does not exist for the large Sagavanirkok River delta area (at approximately 460 km E, 7800 km N) where numerous other bear dens have been identified (see Fig. 3).

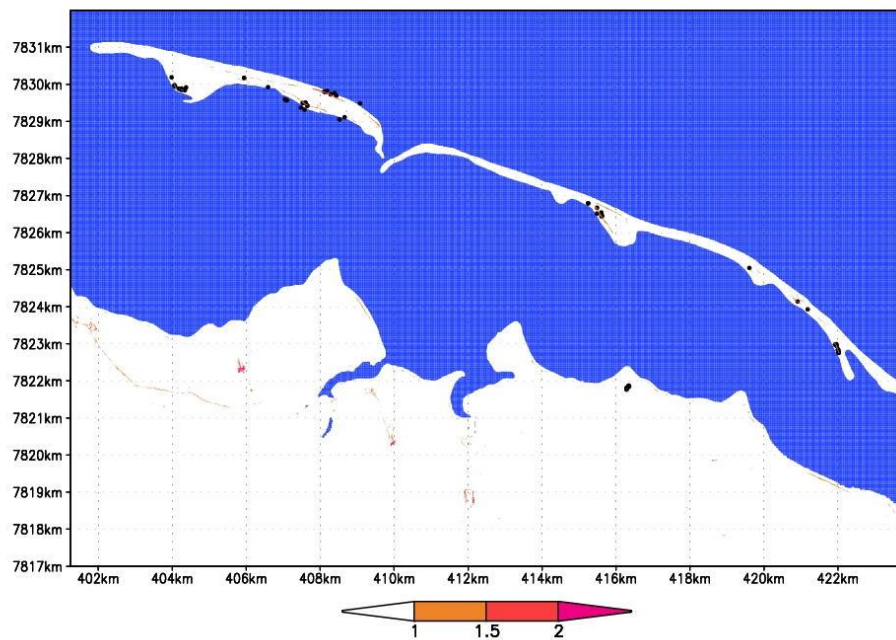


Fig. 20: Equilibrium (maximum possible) snow-drift depths (m) from NE and SW winds for the NW simulation domain. The black dots are historical denning sites.

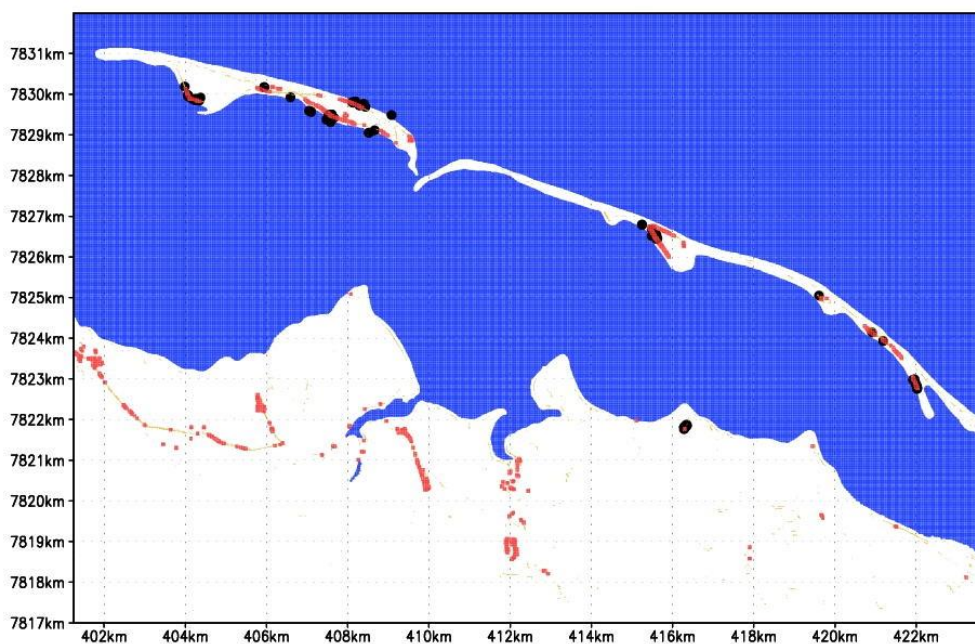


Fig. 21: Snow-drift locations from NE winds for the NW simulation domain. The red grid-cell markers identify drifts deeper than 1 m, and have been increased in total width by 25 m to improve visibility. The lighter colored, thin lines indicate drifts less than 1-m deep. The black dots indicate historical denning sites.

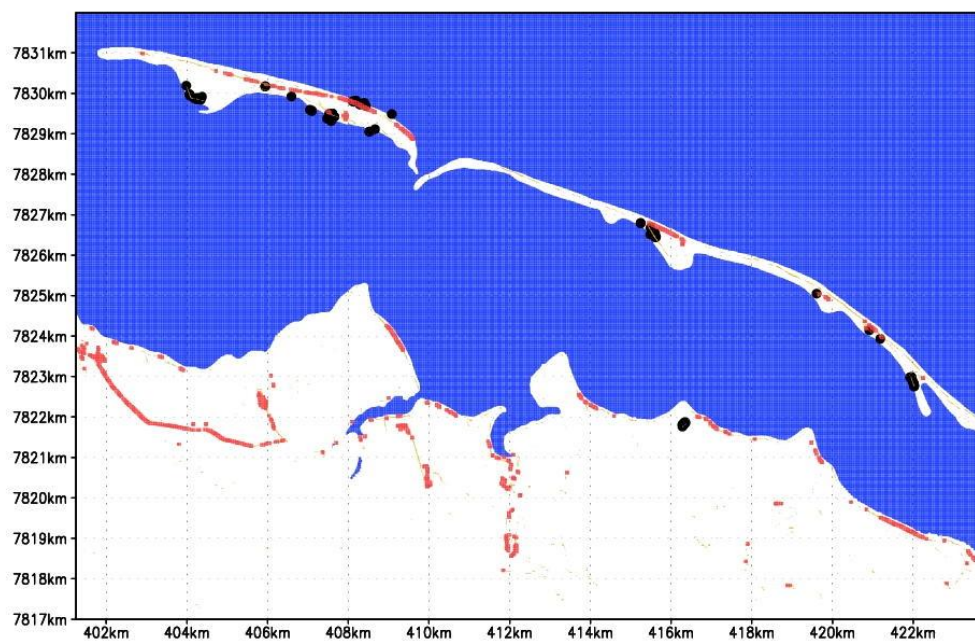


Fig. 22: Similar to Fig. 21, except for SW winds.

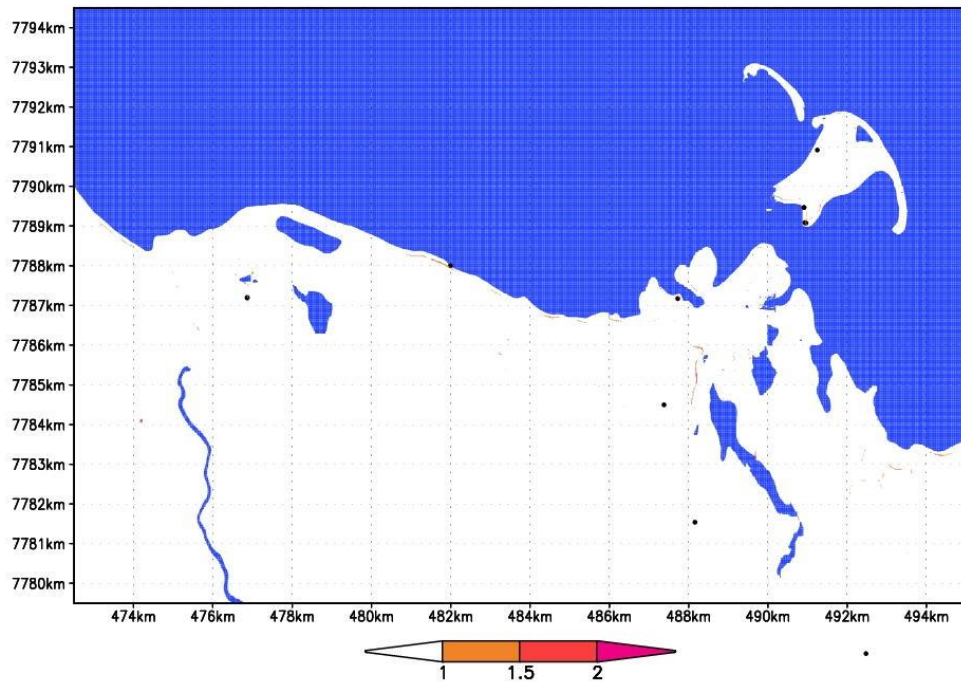


Fig. 23: Equilibrium (maximum possible) snow-drift depths (m) from NE and SW winds for the SE simulation domain. The black dots are historical denning sites.

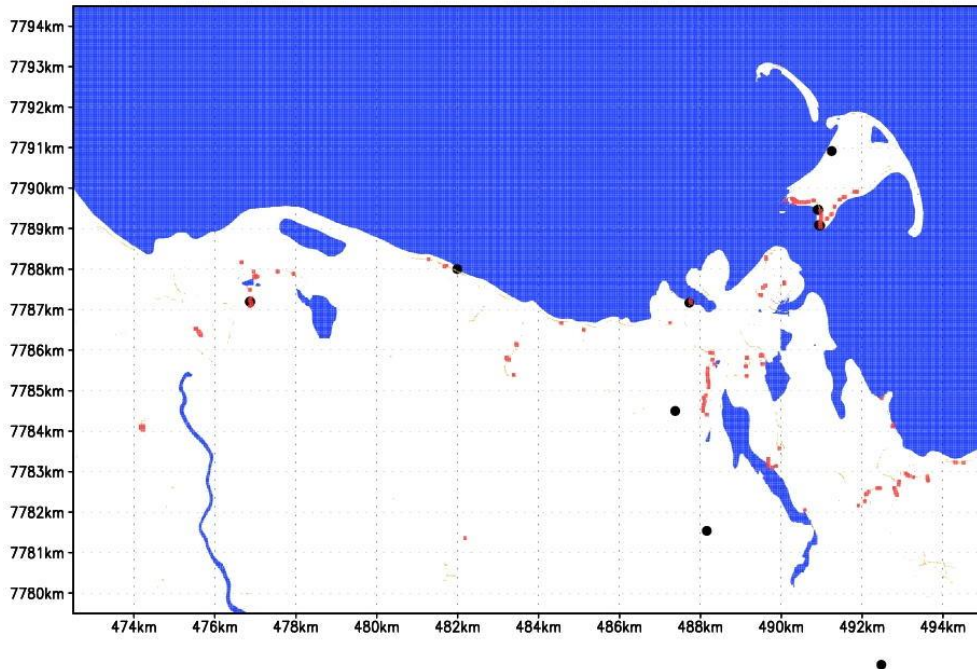


Fig. 24: Snow-drift locations from NE winds for the SE simulation domain. The red grid-cell markers identify drifts deeper than 1 m, and have been increased in total width by 25 m to improve visibility. The lighter colored, thin lines indicate drifts less than 1-m deep. The black dots indicate historical denning sites.

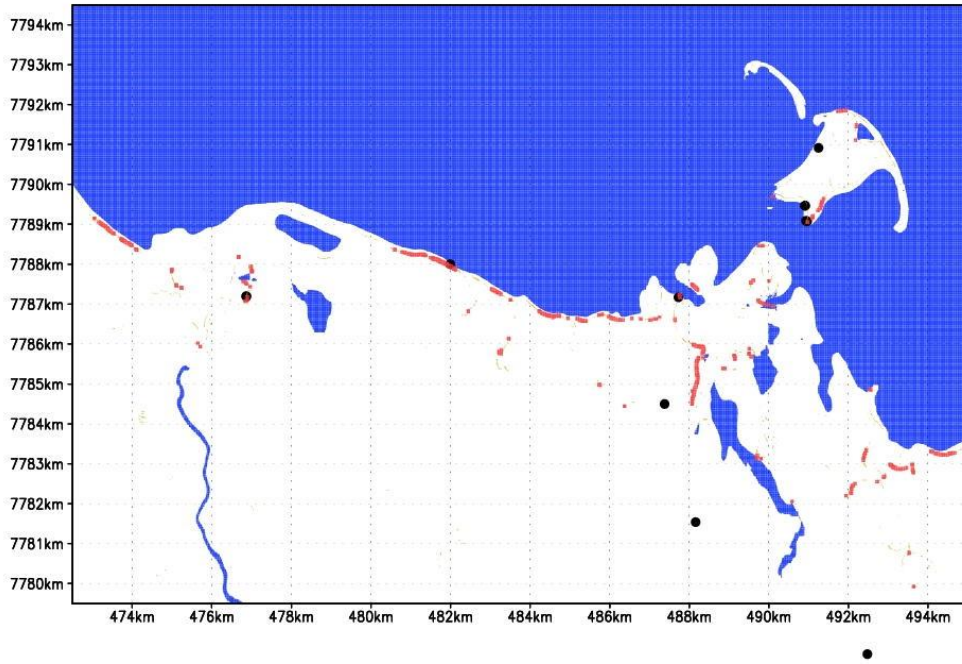


Fig. 25: Similar to Fig. 24, except for SW winds.

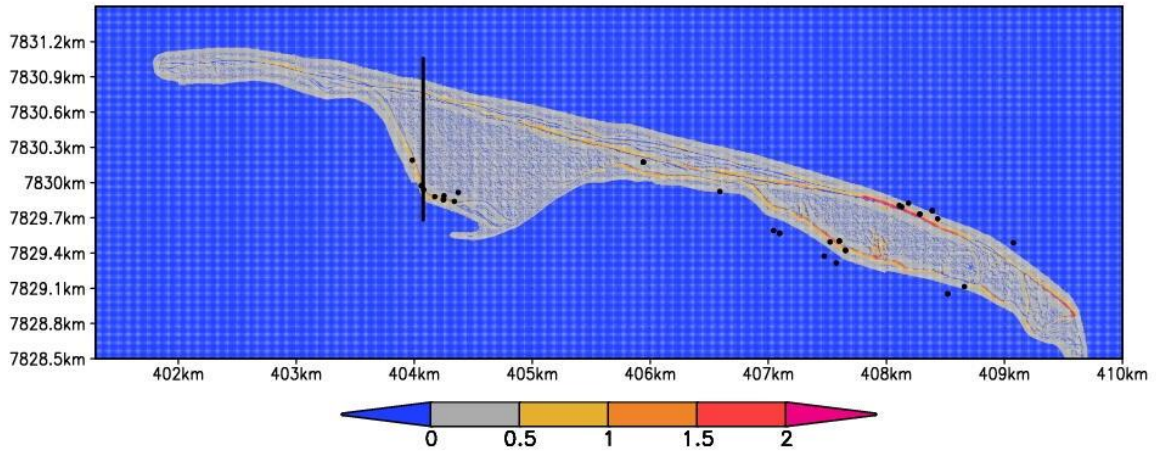


Fig. 26: Zoom-in on Pingok Island from the NW simulation domain, showing equilibrium drift depths (m) from NE and SW winds. Black dots indicate historical denning sites. The line indicates the location of the cross-sectional profile shown in Fig. 27.

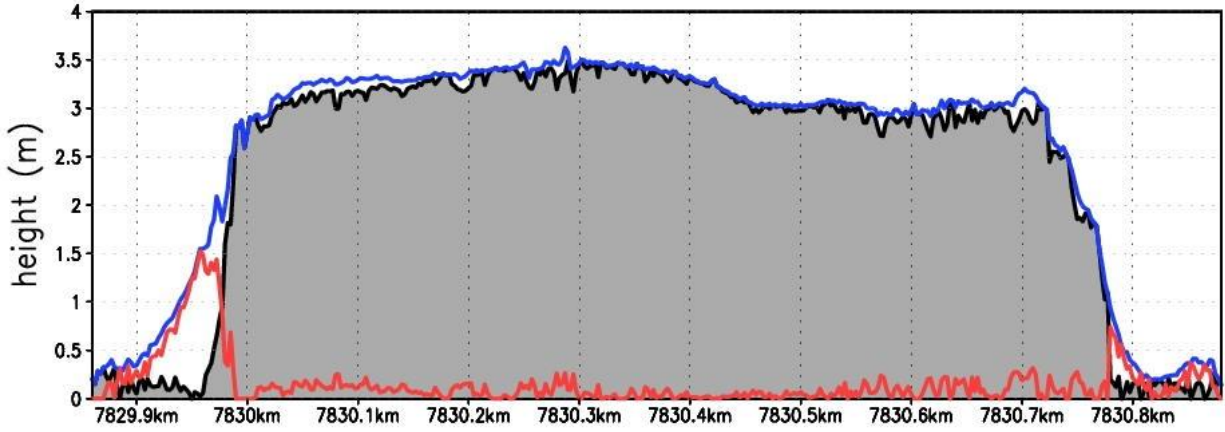


Fig. 27: Cross-sectional profile of the line given in Fig. 26. The black line and grey shading is the land, blue is the equilibrium snow lying over the land, and the red line is the snow-depth profile.

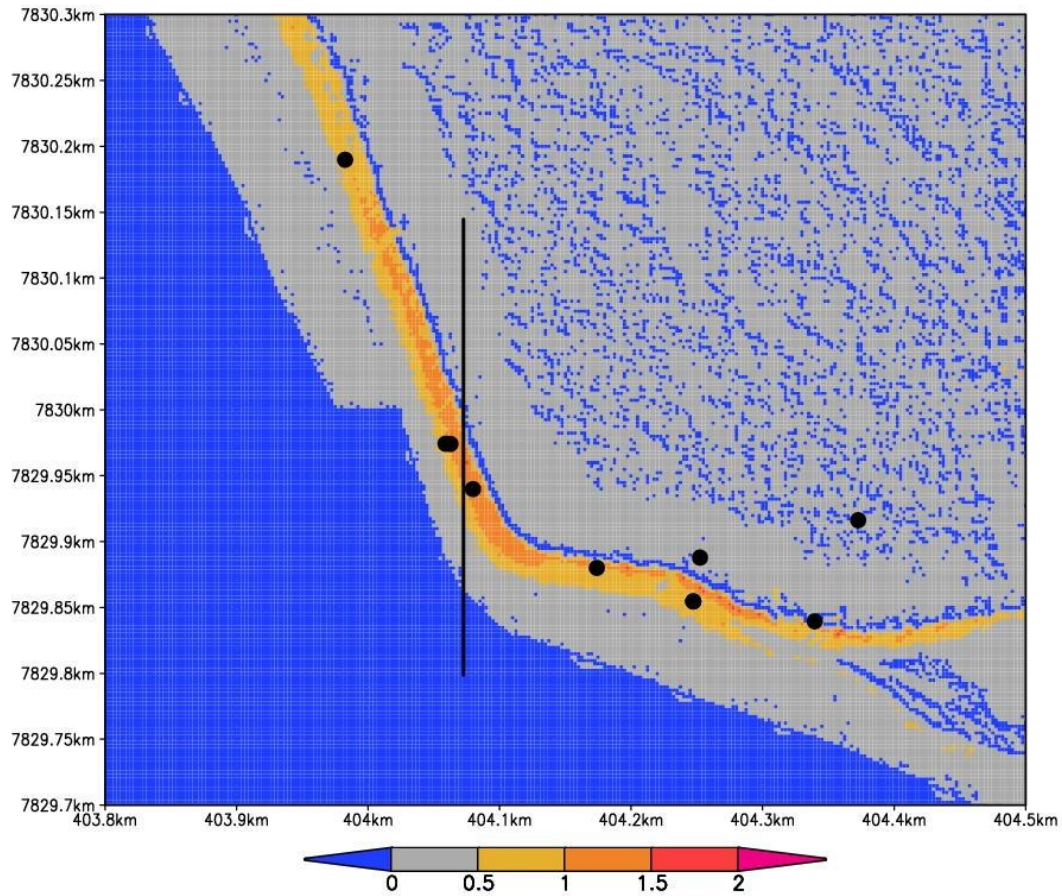


Fig. 28: Zoom-in of Fig. 26. The line indicates the location of the cross-sectional profile shown in Fig. 30. Denning sites outside the drift location are thought to be because of GPS errors.

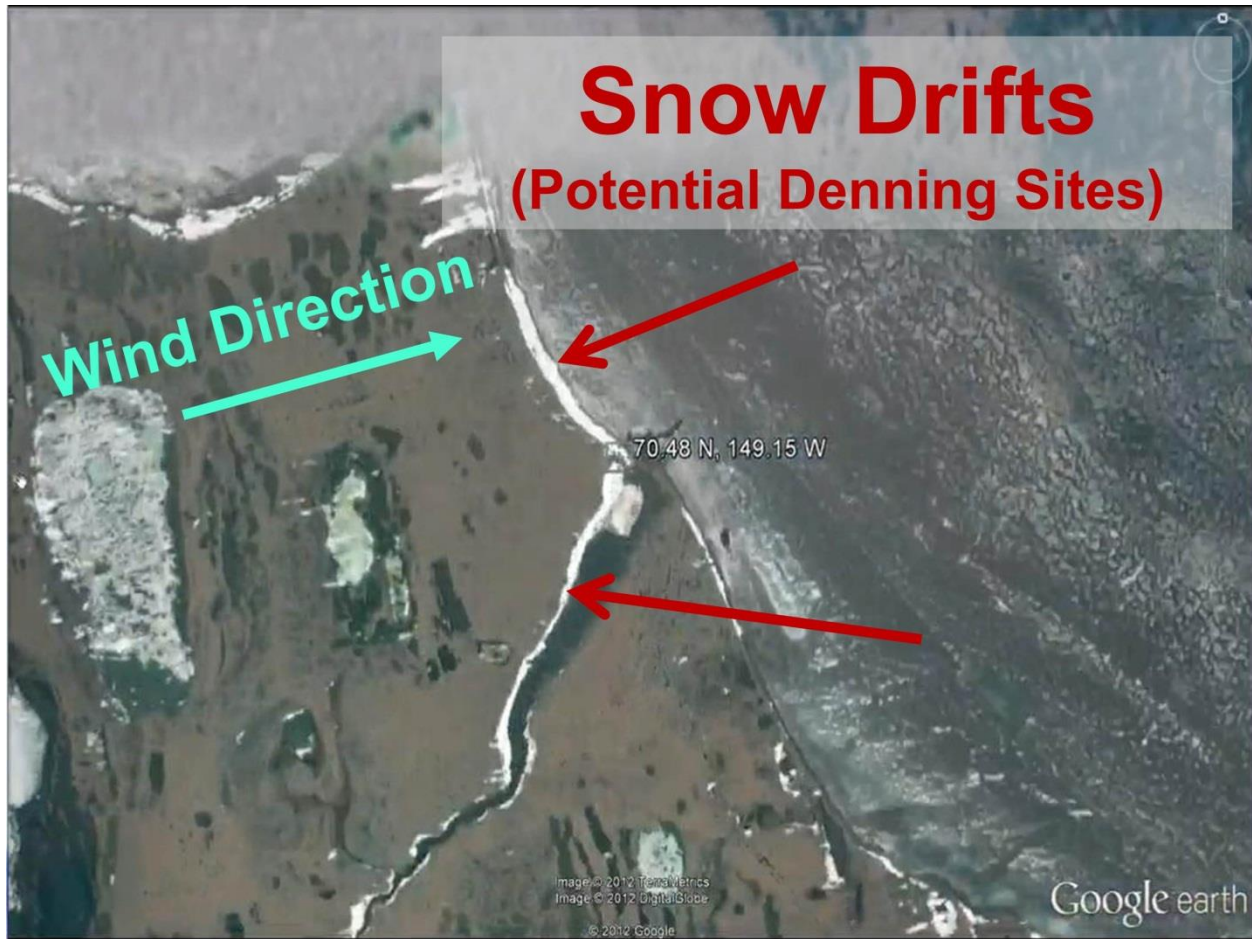


Fig. 29: Google Earth image of a late-season snowdrift along Alaska's Arctic coast, showing the ribbon-like character of the deepest snowdrifts, similar to that simulated by SnowModel in Fig. 28.

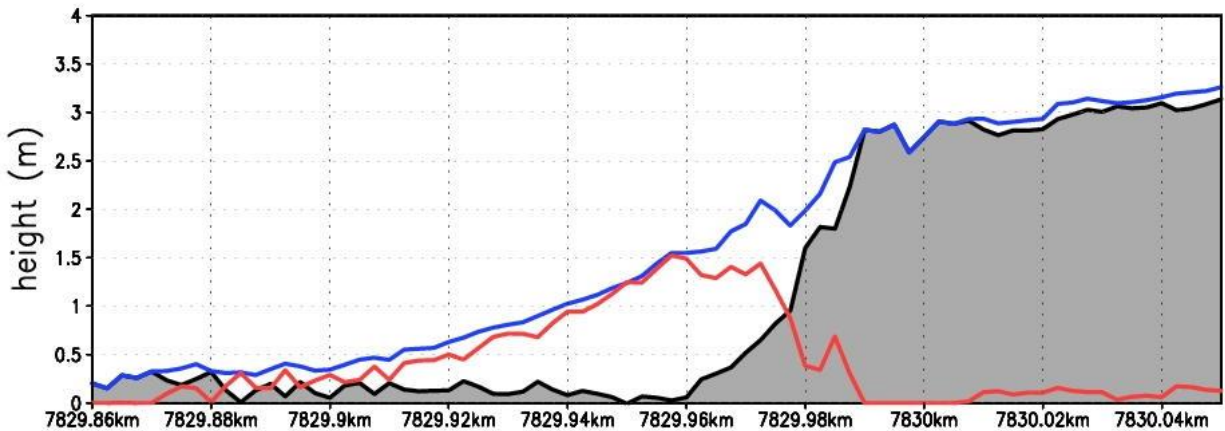


Fig. 30: Cross-sectional profile of the line given in Fig. 28. The black line and grey shading is the land, blue is the equilibrium snow lying over the land, and the red line is the snow-depth profile.

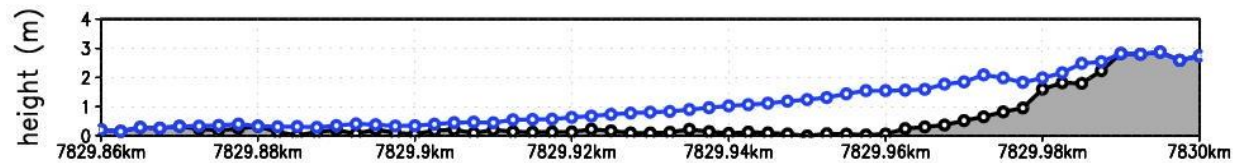


Fig. 31: Zoom-in of the cross-sectional profile of the line given in Fig. 28 and the profile given in Fig. 30. The black line with the open circles and the grey shading are the land, and the blue is the equilibrium snow lying over the land. The marks represent the grid increment (2.5-m). Note the distance along the x axis is 140 m, and the y axis is 4 m. To provide an image of the 1-to-1 aspect ratio found in nature, either the y axis would have to be reduced to $1/4^{\text{th}}$ the current height, or the x axis would have to be 4 times longer; in the natural system these drifts are very long and thin. As shown in Fig. 30, the maximum thickness of this drift is 1.5 m.



Fig. 32: Pictures from the field showing what the general character of the cut-banks and snowdrifts simulated by SnowModel (e.g., Fig. 31) look like in the natural system; they are only a few meters high and they can fill to capacity with the available snow and wind. Note the view is reversed from that of Fig. 31 (in Fig. 31 the wind blew from right to left and the cut-bank faces left, while in this figure the wind blew from left to right and the cut-bank faces right). Photos courtesy of Craig Perham and Dick Shideler.

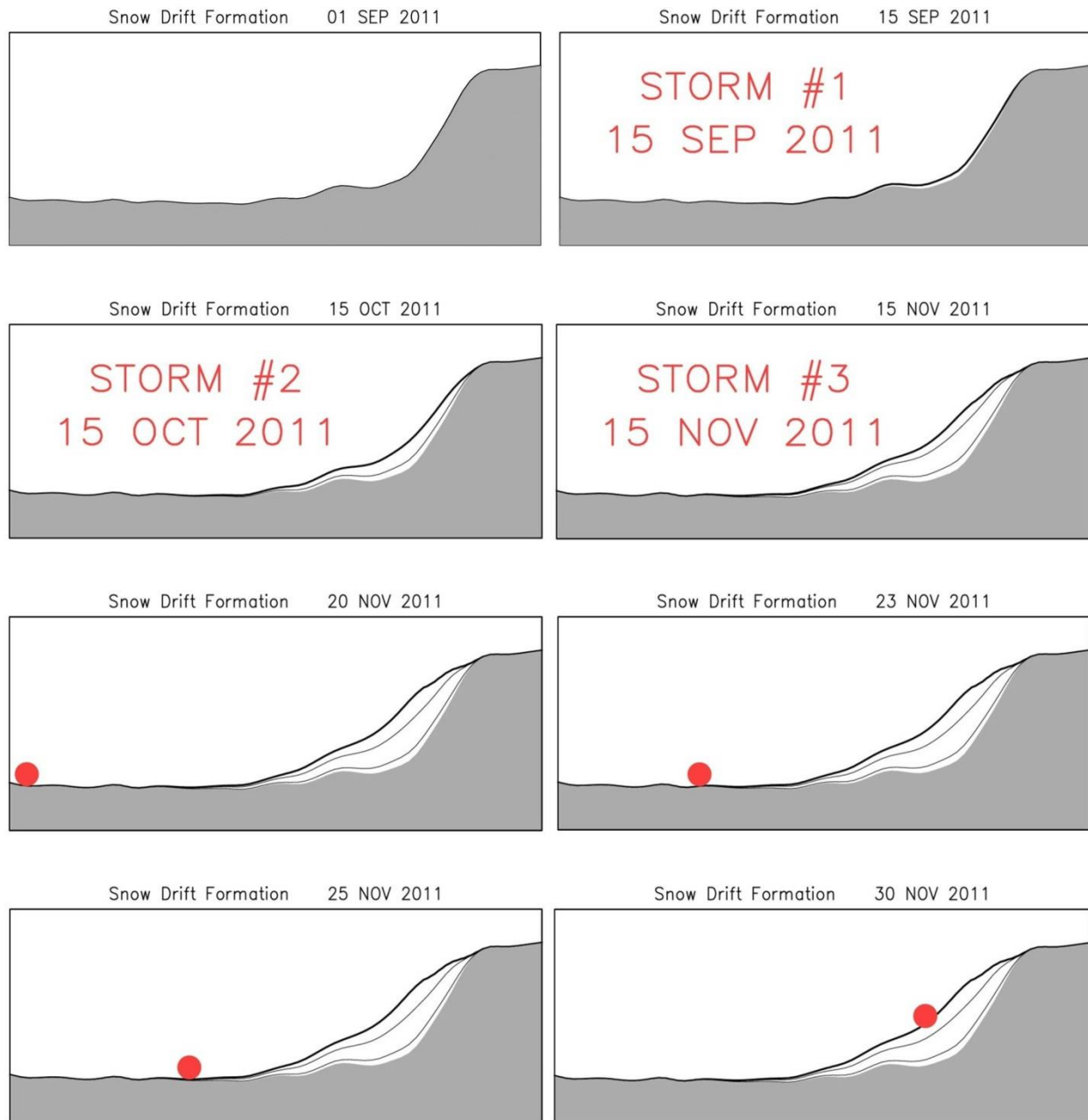


Fig. 33: Excerpts from a 1458-frame movie made from an example SnowModel simulation of snowdrift growth and polar bear denning (display saved every 4 hours from 1 September through 15 April). The simulation assumes a snowstorm on the 15th of every month; the mother polar bear (big red dot) occupied the den (blue circle) on 1 December; she had two cubs (small red dots) on 1 January; the cubs grew during the next few months; and they all emerged from the den in early April.

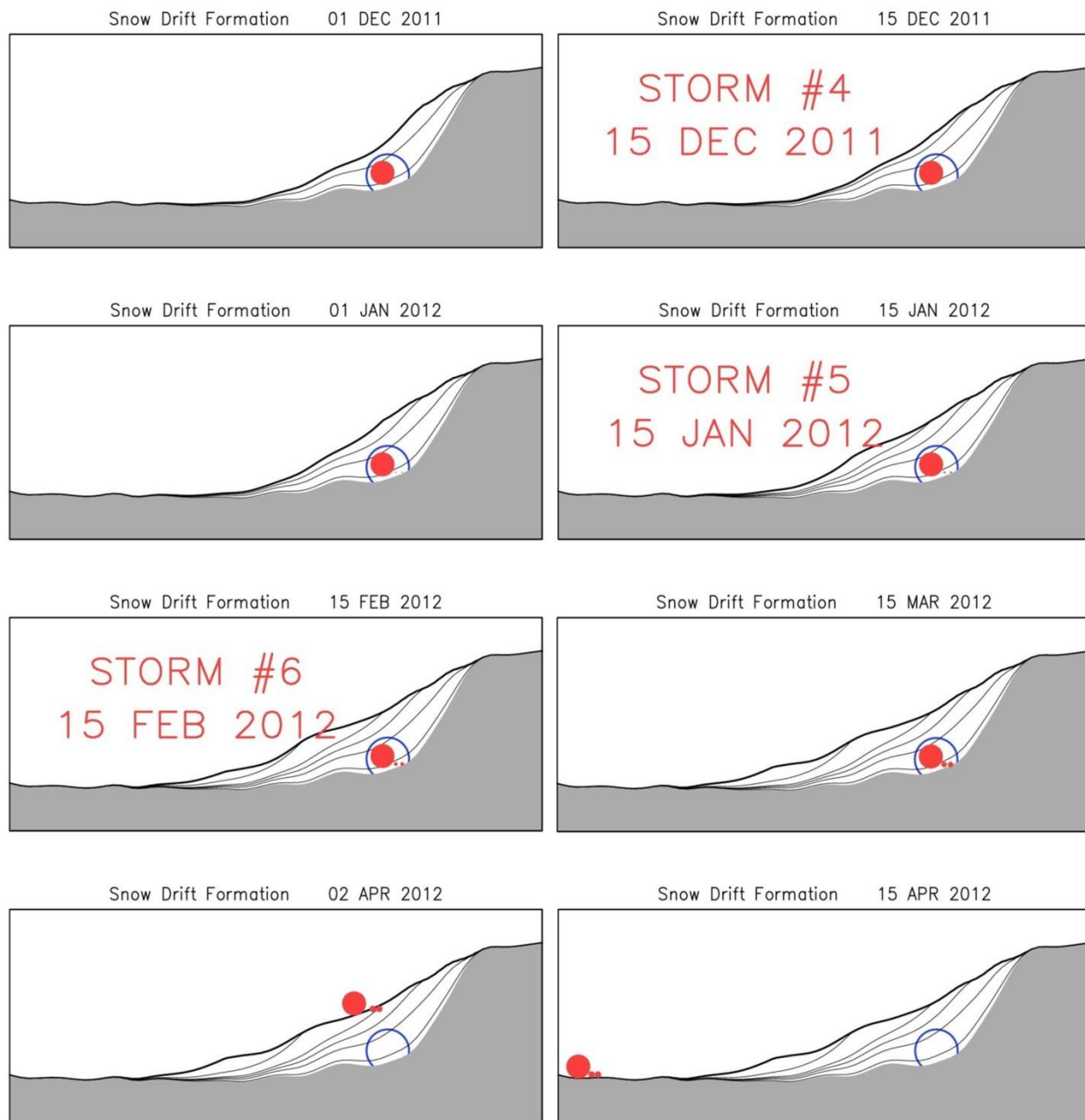


Fig. 33, continued.

Fig. 34: Flashcard front (top) and back (bottom) that the education and outreach lessons (2.8 GB of material) are printed on. These cards are available to education and outreach personnel.



TABLES:

Table 1. SnowModel land-cover classification legend.

Class	Description	Snow-holding depth (m)	Example	General Class
1	coniferous forest	15	spruce-fir/taiga/lodgepole	forest
2	deciduous forest	12	aspen forest	forest
3	mixed forest	14	aspen/spruce-fir/low taiga	forest
4	scattered short-conifer	8	pinyon-juniper	forest
5	clearcut conifer	4	stumps and regenerating	forest
6	mesic upland shrub	0.5	deeper soils, less rocky	shrub
7	xeric upland shrub	0.25	rocky, windblown soils	shrub
8	playa shrubland	1	greasewood, saltbush	shrub
9	shrub wetland/riparian	1.75	willow along streams	shrub
10	erect shrub tundra	0.65	arctic shrubland	shrub
11	low shrub tundra	0.3	low to medium arctic shrubs	shrub
12	grassland rangeland	0.15	graminoids and forbs	grass
13	subalpine meadow	0.25	meadows below treeline	grass
14	tundra (non-tussock)	0.15	alpine, high arctic	grass
15	tundra (tussock)	0.2	graminoid and dwarf shrubs	grass
16	prostrate shrub tundra	0.1	graminoid dominated	grass
17	arctic gram. wetland	0.2	grassy wetlands, wet tundra	grass
18	bare	0.01		bare
19	water/possibly frozen	0.01		water
20	permanent snow/glacier	0.01		water
21	residential/urban	0.01		human
22	tall crops	0.4	e.g., corn stubble	human
23	short crops	0.25	e.g., wheat stubble	human
24	ocean/possibly frozen	0.01		water

Table 2. Meteorological datasets used in project simulations and analyses.

ID Number	Station Name	Source	Elevation (m)	Easting (m)	Northing (m)	Dates
1	Milne Point F-Pad	GW Scientific	2	401065	7824533	2008-2010
2	Cottle Island	GW Scientific	1	422145	7822827	2009-2011
3	Duck Island	GW Scientific	3	462788	7796274	2008-2011
4	Badami	GW Scientific	8	498843	7781207	2009-2010
5	Deadhorse Airport	NOAA-NCDC	18	444145	7787962	1979-2011
6	Prudhoe Bay	NOAA-NCDC	5	443211	7811189	2005-2011
7	Kuparuk Airport	NOAA-NCDC	20	402431	7804870	1991-2011
8	Betty Pingo	UAF	12	428609	7798211	2008-2011
9	Kadleroshilik River	UAF	24	475276	7774146	2008-2010
10	Prudhoe Bay SNOTEL	NRCS	9	440957	7796409	2007-2011
11	NARR grid point	NCEP	0	426244	7811748	1979-2011

Table 3. Example MicroMet and SnowModel parameter and setup file (snowmodel.par).

```

! snowmodel.par file.
!
! Define any required constants specific to this model run.
!
! The following must be true:
!
!   All comment lines start with a ! in the first position.
!
!   Blank lines are permitted.
!
!   All parameter statements start with the parameter name, followed
!   by a space, followed by an =, followed by a space, followed by
!   the actual value, with nothing after that. These statements can
!   have leading blanks, and must fit within 80 columns.
!
!   Also note that all of the input numbers follow standard
!   fortran_77 convention where anything starting with the letters
!   "i" through "n" are integers, and all others are real numbers.

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!  GENERAL MODEL SETUP
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

! Number of x and y cells in the computational grid.
!!! (recommended default value: nx = user input required)
!!! (recommended default value: ny = user input required)
    nx = 9000
    ny = 6000

! deltax = grid increment in x direction. Meters.
! deltay = grid increment in y direction. Meters.
!!! (recommended default value: deltax = user input required)
!!! (recommended default value: deltay = user input required)
    deltax = 2.5
    deltay = 2.5

! Location (like UTM, in meters) value of lower-left grid point.
!   xmn = value of x coordinate in center of lower left grid cell.
!   Meters.
!   ymn = value of y coordinate in center of lower left grid cell.
!   Meters.
!!! (recommended default value: xmn = user input required)
!!! (recommended default value: ymn = user input required)
! Box 1.
    xmn = 401250.
    ymn = 7817000.
! Box 2.

```

```

!      xmn = 472500.
!      ymn = 7779500.

! Model time step, dt.  Should be the same increment as in the input
! data file.  In seconds.
!!! (recommended default value: dt = user input required)
! One day.
!      dt = 86400.0
! Six hours.
!      dt = 21600.0
! Three hours.
!      dt = 10800.0
! One hour.
!      dt = 3600.0

! Start year of input data file.  Four digit year.  Integer.
!!! (recommended default value: iyear_init = user input required)
!      iyear_init = 2010

! Start month of input data file.  One or two digit month.  Integer.
!!! (recommended default value: imonth_init = user input required)
!      imonth_init = 9

! Start day of input data file.  One or two digit day.  Integer.
!!! (recommended default value: iday_init = user input required)
!      iday_init = 1

! Start hour of input data file.  Local time, like in solar time.
! Decimal hour.  Each day the clock runs from 0.00 through 23.99.
! Real.
!!! (recommended default value: xhour_init = user input required)
!      xhour_init = 1.0
!      xhour_init = 12.0

! Number of model iterations defines how many times to process.
!!! (recommended default value: max_iter = user input required)
!      max_iter = 304
!      max_iter = 1

! MicroMet requires a met input file that has data in a specific
! format (see the MicroMet documentation).  For the case of
! inputs from more than a single station, the input met file
! requires an initial number indicating the number of valid
! stations at a given time step.  For the case of a single
! station, this number can be dropped (it doesn't have to be
! though).  Identify whether to process as a single station
! without the valid station count (isingle_stn_flag = 1), or
! or to process with the count included (for multi stations
! and single stations with the number of stations included;
! isingle_stn_flag = 0).
!!! (recommended default value: isingle_stn_flag = user input
!!! required)

```

```

    isingle_stn_flag = 0

! For the case of isingle_stn_flag = 1, the input file can be a GrADS
!   binary file.  Identify whether that is the case (=1, else =0).
!!! (recommended default value: igrads_metfile = 0)
    igrads_metfile = 0

! Define the meteorological input file name.  Note that this input
!   file has very specific input format requirements (see the
!   MicroMet preprocessor).  The required met variables are:
!   Tair (deg C or K), rh (%), wind speed (m/s), wind direction
!   (0-360 True N), and precipitation (mm/time_step).
!!! (recommended default value: met_input_fname = user input required)
!   met_input_fname = met/met_station_408_dyly.dat
!   met_input_fname = met/met_station_408_hrly.dat
    met_input_fname = ../met/mm/daily/2010-2011/ak_met_2010-
2011_daily.dat

! The number used as an undefined value for both inputs and outputs.
!!! (recommended default value: undef = -9999.0)
    undef = -9999.0

! Define whether the topography and vegetation input files will
!   be ARC/INFO ascii text (grid) files, or a GrADS binary file
!   (ascii = 1.0, GrADS = 0.0).
!!! (recommended default value: ascii_topoveg = 1.0)
    ascii_topoveg = 0.0

! Define the GrADS topography and vegetation input file name
!   (record 1 = topo, record 2 = veg).  Note that if you are using
!   ascii files, you still cannot comment this line out (it is okay
!   for it to point to something that doesn't exist) or you will
!   an error message.
!!! (recommended default value: topoveg_fname = xxxxxx)
    topoveg_fname = topo_veg/box1_topo_veg_2.5m.gdat
!   topoveg_fname = topo_veg/box2_topo_veg_2.5m.gdat

! For the case of using ascii text topography and vegetation files,
!   provide the file names.  Note that if you are using a grads
!   file, you still cannot comment these two lines out (it is okay
!   for it to point to something that doesn't exist) or you will
!   an error message.
!!! (recommended default value: topo_ascii_fname = user input
required)
!!! (recommended default value: veg_ascii_fname = user input required)
    topo_ascii_fname = xxxxxx
    veg_ascii_fname = xxxxxx

! The vegetation types are assumed to range from 1 through 30.  The
!   last 6 types are available to be user-defined vegetation types
!   and vegetation snow-holding depths.  The first 24 vegetation
!   types, and the associated vegetation snow-holding depths

```

```

!   (meters), are hard-coded to be:
!
! code description          veg_shd  example          class
!
! 1  coniferous forest      15.00  spruce-fir/taiga/lodgepole
forest
! 2  deciduous forest       12.00  aspen forest
forest
! 3  mixed forest           14.00  aspen/spruce-fir/low taiga
forest
! 4  scattered short-conifer 8.00  pinyon-juniper
forest
! 5  clearcut conifer       4.00  stumps and regenerating
forest
!
! 6  mesic upland shrub     0.50  deeper soils, less rocky  shrub
! 7  xeric upland shrub     0.25  rocky, windblown soils   shrub
! 8  playa shrubland        1.00  greasewood, saltbush     shrub
! 9  shrub wetland/riparian 1.75  willow along streams     shrub
! 10 erect shrub tundra     0.65  arctic shrubland         shrub
! 11 low shrub tundra       0.30  low to medium arctic shrubs shrub
!
! 12 grassland rangeland    0.15  graminoids and forbs     grass
! 13 subalpine meadow       0.25  meadows below treeline   grass
! 14 tundra (non-tussock)   0.15  alpine, high arctic      grass
! 15 tundra (tussock)       0.20  graminoid and dwarf shrubs grass
! 16 prostrate shrub tundra 0.10  graminoid dominated      grass
! 17 arctic gram. wetland   0.20  grassy wetlands, wet tundra grass
!
! 18 bare                   0.01  bare                      bare
!
! 19 water/possibly frozen  0.01  water                     water
! 20 permanent snow/glacier 0.01  water                     water
!
! 21 residential/urban      0.01  human                     human
! 22 tall crops             0.40  e.g., corn stubble        human
! 23 short crops            0.25  e.g., wheat stubble       human
! 24 ocean                  0.01  water                     water
!
! 25 user defined (see below)
! 26 user defined (see below)
! 27 user defined (see below)
! 28 user defined (see below)
! 29 user defined (see below)
! 30 user defined (see below)
!
! Define the vegetation snow-holding depths (in meters) for each
! of the user-defined vegetation types. The numbers in the
! list below correspond to the vegetation-type numbers in the
! vegetation-type data array (veg type 25.0 -> veg_shd_25). Note
! that even if these are not used, they cannot be commented out
! or you will get an error message.

```

```

!!! (recommended default value: veg_shd_25 = 0.10)
!!! (recommended default value: veg_shd_26 = 0.10)
!!! (recommended default value: veg_shd_27 = 0.10)
!!! (recommended default value: veg_shd_28 = 0.10)
!!! (recommended default value: veg_shd_29 = 0.10)
!!! (recommended default value: veg_shd_30 = 0.10)
    veg_shd_25 = 0.10
    veg_shd_26 = 0.10
    veg_shd_27 = 0.10
    veg_shd_28 = 0.10
    veg_shd_29 = 0.10
    veg_shd_30 = 0.10

! Define whether the vegetation will be constant or defined by the
! topography/vegetation input file name (0.0 means use the file,
! 1.0 or greater means use a constant vegetation type equal to
! the number that is used). This will define the associated
! veg_shd that will be used. The reason you might use a constant
! vegetation type is to avoid generating a veg-distribution file.
!!! (recommended default value: const_veg_flag = 0.0)
!     const_veg_flag = 12.0
!     const_veg_flag = 0.0
!     const_veg_flag = 1.0

! For the case where you have vegetation height data, and want to
! use that instead of the tables above, you must also provide a
! file that defines the vegetation height (cm) of each grid cell.
! iveg_ht_flag = 0 means the file is not provided and the required
! array will be generated by filling it with the values listed in
! the above vegetation summary. iveg_ht_flag = -1 means the data
! file is a grads binary file with the name veg_ht.gdat.
! iveg_ht_flag = 1 means the data file is an ascii text file with
! the same format that ascii topo and veg files would have and with
! a name of veg_ht.asc. Note that the .gdat and .asc files are
! y-reversed (see ascii topo/veg notes). It is assumed that the
! file, if provided, is located in an 'topo_veg/' directory off
! of the main model directory. Note that this height information
! is primarily used as the vegetation snow-holding depth, thus the
! model is expecting average vegetation heights for each grid cell,
! not maximum heights.
!!! (recommended default value: iveg_ht_flag = 0)
    iveg_ht_flag = 0

! The latitude of domain center (decimal degrees). This only needs
! to be approximate, since it is used to define some of the solar
! radiation calculations.
!!! (recommended default value: xlat = user input required)
    xlat = 70.2

! For the case where your domain spans enough latitude that there
! is significant variation in solar radiation from the top to
! the bottom of the domain, you must also provide a file that

```

```

!   defines the latitude (in decimal degrees) of the center of
!   each grid cell.  lat_solar_flag = 0 means the file is not
!   provided and the required array will be generated by filling
!   it with the constant xlat value listed above.  lat_solar_flag
!   = -1 means the data file is a grads binary file with the name
!   grid_lat.gdat.  lat_solar_flag = 1 means the data file is an
!   ascii text file with the same format that ascii topo and veg
!   files would have and with a name of grid_lat.asc.  Note that
!   the .gdat and .asc files are y-reversed (see ascii topo/veg
!   notes).  It is assumed that the file, if provided, is located
!   in an 'extra_met/' directory off of the main model directory.
!!! (recommended default value: lat_solar_flag = 0)
    lat_solar_flag = 0

! For the case where your domain spans enough longitude that there
! is significant variation in solar radiation from the side to
! the side of the domain, you must also provide a file that
! defines the longitude (in decimal degrees) of the center of
! each grid cell.  In this case it also no longer makes sense
! to be running the model in local time, and UTC (or GMT) time
! is required.  UTC_flag = 0.0 means the the model will be run
! in local time.  UTC_flag = -1.0 means the longitude data file
! is a grads binary file with the name grid_lon.gdat.  UTC_flag
! = 1 means the longitude data file is an ascii text file with
! the same format that ascii topo and veg files would have and
! with a name of grid_lon.asc.  Note that the .gdat and .asc
! files are y-reversed from each other (see ascii topo/veg
! notes).  It is assumed that the file, if provided, is located
! in an 'extra_met/' directory off of the main model directory.
! Note that if UTC_flag is non-zero, lat_solar_flag should
! probably also be non-zero.
!!! (recommended default value: UTC_flag = 0.0)
    UTC_flag = 0.0

! Define which models that are going to run.  A value of 1.0 means
! that you want to run the model (0.0 means don't run it).
! Currently the model is only configured to run the following
! combinations (micromet), (micromet, enbal), (micromet, enbal,
! snowpack), (micromet, snowtran), or (micromet, enbal, snowpack,
! snowtran).
!!! (recommended default value: run_micromet = user input required)
!!! (recommended default value: run_enbal = user input required)
!!! (recommended default value: run_snowpack = user input required)
!!! (recommended default value: run_snowtran = user input required)
    run_micromet = 1.0
    run_enbal = 1.0
    run_snowpack = 1.0
    run_snowtran = 1.0

! Define whether you are doing a standard model simulation, or a
! precipitation and/or melt correction simulation to force the
! model towards observed swe distributions.  For a standard

```

```

! simulation, use irun_corr_factor = 0, for a precipitation/melt
! factor correction run, use irun_corr_factor = 1. Note that
! irun_corr_factor = 1 requires swe data and some extra input
! files. See the dataassim_user.f subroutine for details. If
! you are using this feature, you will also need to edit the
! outputs_user.f file.
!!! (recommended default value: irun_corr_factor = user input
!!! required)
!       irun_corr_factor = 0
!       irun_corr_factor = 1

! Define whether want to implement the history restart option.
! This creates periodic output files that allow you to restart
! the simulation from somewhere other than the beginning. All
! of the required data arrays are saved to a grads file that is
! read back in when the simulation is restarted. The new
! simulation starts at the history restart iteration, with the
! initial conditions defined by the saved data arrays. The
! new simulation starts in the micromet file where the original
! simulation ended. This is also true of the output files,
! if the run has been set up to write continuously to a given
! output file. This option also assumes that you have not made
! any changes to the model setup; it does no error checking.
! ihrestart_flag = -2 turns this option off, ihrestart_flag = -1
! does the restart saves, and ihrestart_flag >= 0 runs a history
! restart where the number defines the restart iteration (e.g.,
! ihrestart = 720 will set iteration 720 as the initial
! condition for the start of the new simulation. For the case
! where ihrestart_flag = 0, and i_dataassim_loop <= -1, the
! run will start at the beginning of the second data assimilation
! loop.
!!! (recommended default value: ihrestart_flag = -2)
!       ihrestart_flag = -2
!       ihrestart_flag = -1
!       ihrestart_flag = 3444
!       ihrestart_flag = 0

! For the case where you are running a history restart, define
! whether you want to restart the run during a standard run or
! the first loop of a data assimilation model run
! (i_dataassim_loop = 1) or during the second loop of a data
! assimilation run (i_dataassim_loop < 0). The value of this
! negative number is used to define how many observation dates
! you have in your data assimilation run (e.g., -3 would mean
! you have 3 observation dates and nobs_dates = 3 in
! dataassim_user.f). Also note that to correctly run the history
! restart function for the case of a data assimilation run, you
! need to first set i_corr_factor = 0 and save a history restart
! at the end of that run before going on to the second half of
! the assimilation run.
!!! (recommended default value: i_dataassim_loop = 1)
!       i_dataassim_loop = 1

```

```

!      i_dataassim_loop = -1

! If ihrestart_flag = -1, then define how often the restart
! files are generated. This is in iteration units (e.g., every
! 30 days for the model running at hourly time steps would be
! ihrestart_inc = 720). The history restart files will be
! placed in an 'hrestart/' directory off of the main model
! directory. In addition to this time increment, if the
! ihrestart_flag = -1, the last model iteration is always saved
! (so if you just want to save the last iteration, just set
! ihrestart_inc to a number bigger than your maximum iteration).
! Note that if you want to do history restarts with data
! assimilation this increment must be evenly divisible into
! the maximum interation; in other words, the simulation must
! at least do a write at the end of the first data assimilation
! loop.
!!! (recommended default value: ihrestart_inc = 0)
      ihrestart_inc = 0

! The code is set up to write out an individual data file for each
! sub-model (see the sub-model sections below). In addition, you
! can use a user-defined subroutine to output data in the format
! of your choice. Define whether you want the data written out
! to this file (print_user = 1.0, else 0.0). The name and
! location of the user-defined output file(s) will be defined
! within the subroutine. If you are using this feature, you
! will need to edit the outputs_user.f file.
!!! (recommended default value: print_user = 0.0)
      print_user = 1.0

! For printing to the standard output files, define the iteration
! increment for each write to the files (e.g., iprint_inc = 1
! gives every time step, iprint_inc = 24 gives a write at the end
! of each day when using hourly time steps, etc.).
!!! (recommended default value: iprint_inc = 1)
!      iprint_inc = 24
!      iprint_inc = 1

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!! MICROMET MODEL SETUP
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

! Define which fields you want to process/distribute. 1 = do it,
! 0 = don't do it. As an example, if you are running SnowTran-3D
! with no melting, you don't need solar radiation. Also, MicroMet
! provides the ability to to read in an alternate wind dataset
! that has been previously generated with another program, like
! winds from NUATMOS. To do that set the flag to -1. See
! i_wind_flag = -1 in the code for an example of how to do this.
!!! (recommended default value: i_tair_flag = 1)

```

```

!!! (recommended default value: i_rh_flag = 1)
!!! (recommended default value: i_wind_flag = 1)
!!! (recommended default value: i_solar_flag = 1)
!!! (recommended default value: i_longwave_flag = 1)
!!! (recommended default value: i_prec_flag = 1)
    i_tair_flag = 1
    i_rh_flag = 1
    i_wind_flag = 1
    i_solar_flag = 1
    i_longwave_flag = 1
    i_prec_flag = 1

! Force the model to determine a value for every grid cell (=1), or
! just define values for some specified "radius of influence" (=0).
!!! (recommended default value: ifill = 1)
    ifill = 1

! Let the model determine an appropriate "radius of influence" (=0),
! or define the "radius of influence" you want the model to use
(=1).
! 1=use obs interval below, 0=use model generated interval.
!!! (recommended default value: iobsint = 0)
    iobsint = 0

! The "radius of influence" or "observation interval" you want the
! model to use for the interpolation. In units of deltax, deltax.
!!! (recommended default value: dn = 1.0)
    dn = 1.0

! The barnes station interpolation can be done two different ways:
! First, barnes_oi does the interpolation by processing all of
! the available station data for each model grid cell.
! Second, barnes_oi_ij does the interpolation by processing only
! the "n_stns_used" number of stations for each model grid cell.
! For small domains, with relatively few met stations (100's),
! the first way is best. For large domains (like the
! United States, Globe, Pan-Arctic, North America, Greenland)
! and many met stations (like 1000's), the second approach is the
! most efficient. But, the second approach carries the following
! restrictions: 1) there can be no missing data for the fields of
! interest; 2) there can be no missing stations (all stations
! must exist throughout the simulation period); and 3) the
! station met file must list the stations in the same order for
! all time steps. This second method works well for regridding
! atmospheric analyses/model datasets. Also, the MicroMet
! preprocessor can be used to fill in missing data segments.
! Use barnes_lg_domain = 0.0 for the first method, and
! barnes_lg_domain = 1.0 for the second method.
!!! (recommended default value: barnes_lg_domain = 0.0)
    barnes_lg_domain = 0.0

! For the case where barnes_lg_domain = 1.0, define the number

```

```

!   of nearest stations to be used in the interpolation (5 or
!   less).  If barnes_lg_domain = 0.0, n_stns_used is not used, but
!   n_stns_used still needs to have some value.
!!! (recommended default value: n_stns_used = 5)
    n_stns_used = 5

! The curvature is used as part of the wind model.  Define a length
!   scale that the curvature calculation will be performed on.  This
!   has units of meters, and should be approximately one-half the
!   wavelength of the topographic features within the domain.
!!! (recommended default value: curve_len_scale = 500.0)
    curve_len_scale = 10.0

! The curvature and wind_slope values range between -0.5 and +0.5.
!   Valid slopewt and curvewt values are between 0 and 1, with
!   values of 0.5 giving approximately equal weight to slope and
!   curvature.  I suggest that slopewt and curvewt be set such
!   that slopewt + curvewt = 1.0.  This will limit the total
!   wind weight to between 0.5 and 1.5 (but this is not required).
!!! (recommended default value: slopewt = 0.58)
!!! (recommended default value: curvewt = 0.42)
    slopewt = 0.58
    curvewt = 0.42
!   slopewt = 0.25
!   curvewt = 0.75

! Avoid problems of zero (low) winds (for example, turbulence
!   theory, log wind profile, etc., says that we must have some
!   wind.  Thus, some equations blow up when the wind speed gets
!   very small).  This number defines the value that any wind speed
!   below this gets set to.
!!! (recommended default value: windspd_min = 1.0)
    windspd_min = 1.0

! Define whether the model is to use the default monthly lapse
!   rates (= 0) or user supplied monthly lapse rates (= 1).  To use
!   user supplied lapse rates, you have to edit the user lapse rate
!   data array in micromet_code.f (subroutine get_lapse_rates).
!   Note that this implementation currently only defines
!   average-monthly lapse rates.
!!! (recommended default value: lapse_rate_user_flag = 0)
    lapse_rate_user_flag = 0

! Define whether the precipitation adjustment factor, with units of
!   km-1 (kind of a precipitation lapse rate, used to adjust the
!   precipitation for locations above and below the precipitation
!   observing station(s)), is to use the default monthly lapse
!   rates (= 0) or user supplied monthly lapse rates (= 1).  To use
!   user supplied lapse rates, you have to edit the user lapse rate
!   data array in micromet_code.f (subroutine get_lapse_rates).
!!! (recommended default value: iprecip_lapse_rate_user_flag = 0)
    iprecip_lapse_rate_user_flag = 0

```

```

! Define whether you want the model to calculate, use, and output
!   sub-forest-canopy windspeed, incoming solar and longwave
!   radiation (= 1.0), or above canopy values (= 0.0).
!!! (recommended default value: calc_subcanopy_met = 1.0)
    calc_subcanopy_met = 1.0

! Define the canopy gap fraction (0-1). This parameter accounts
!   for solar radiation reaching the snow surface below the canopy,
!   beyond that defined by the canopy transmissivity calculation.
!   In effect, it allows additional solar radiation to penetrate
!   the canopy (e.g., through gaps in the forest), thus increasing
!   melt rates, etc. A gap_frac = 0.0 produces the sub-canopy
!   solar radiation using the default transmissivity calculation,
!   a gap_frac = 1.0 (all gaps) produces sub_canopy radiation equal
!   to the top-of-canopy radiation. If you want the snow in the
!   forest to melt faster, increasing this value will do it.
!!! (recommended default value: gap_frac = 0.2)
    gap_frac = 0.2

! To handle the case, for example, of a met station located in
!   an inversion layer recording high relative humidity, and the
!   model producing anomalously high cloud-cover fractions, the
!   cloud_frac_factor can be used to decrease the simulated cloud
!   fraction. This number is multiplied by the calculated cloud
!   fraction. For example, a cloud_frac_factor = 1.0 produces the
!   simulated cloud fraction, a cloud_frac_factor = 0.5 produces
!   the half the simulated cloud fraction, and a cloud_frac_factor
!   = 0.0 forces the simulated cloud fraction to be zero.
!!! (recommended default value: cloud_frac_factor = 1.0)
    cloud_frac_factor = 1.0

! Define whether the simulation will assimilate shortwave radiation
!   observations (no = 0.0, yes = 1.0). If yes, the model assumes
!   there is a shortwave radiation station data file, called
!   shortwave.dat, in an 'extra_met/' directory off of the main
!   model directory. See the micromet code for file format details.
!!! (recommended default value: use_shortwave_obs = 0.0)
    use_shortwave_obs = 0.0

! Define whether the simulation will assimilate longwave radiation
!   observations (no = 0.0, yes = 1.0). If yes, the model assumes
!   there is a longwave radiation station data file, called
!   longwave.dat, in an 'extra_met/' directory off of the main model
!   directory. See the micromet code for file format details.
!!! (recommended default value: use_longwave_obs = 0.0)
    use_longwave_obs = 0.0

! Define whether the simulation will assimilate surface pressure
!   observations (no = 0.0, yes = 1.0). If yes, the model assumes
!   there is a surface pressure station data file, called
!   sfc_pressure.dat, in an 'extra_met/' directory off of the main

```

```

!   model directory.  See the micromet code for file format details.
!!! (recommended default value: use_sfc_pressure_obs = 0.0)
    use_sfc_pressure_obs = 0.0

! The code is set up to write out an individual data file for each
!   sub-model.  Define whether you want the data written out
!   (print_micromet = 1.0, else 0.0), and the name of that output
!   file.
!!! (recommended default value: print_micromet = 1.0)
!!! (recommended default value: micromet_output_fname =
!!! outputs/micromet.gdat)
    print_micromet = 0.0
    micromet_output_fname = outputs/micromet.gdat

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!  SNOWTRAN-3D MODEL SETUP
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

! Define whether the threshold surface shear velocity will be
!   constant during the simulation (Utau_t_flag = 0.0), or whether
!   it will evolve as a function of air temperature and wind speed
!   (Utau_t_flag = 1.0).
!!! (recommended default value: Utau_t_flag = 0.0)
    Utau_t_flag = 0.0
!   Utau_t_flag = 1.0

! For the case of Utau_t_flag = 0.0, define what the threshold
!   surface shear velocity (m/s) will be during the simulation.
!!! (recommended default value: Utau_t_const = 0.25)
    Utau_t_const = 0.25

! Define whether you want to apply the Tabler subgrid algorithm.
!   Flag = 0.0 is the original model (no subgrid redistribution, as
!   in the original model), flag = 1.0 uses Tabler (1975) to define
!   equilibrium profiles for topographic drift catchments.  Note
!   that it is not recommended to use this (= 1.0) except under
!   special situations.
!!! (recommended default value: subgrid_flag = 0.0)
    subgrid_flag = 0.0
!   subgrid_flag = 1.0

! The implementation of Tabler (1975) allows the user to define
!   the area immediately upwind of any Tabler drift trap to be an
!   erosion area where no snow is allowed to accumulate.  This area
!   is defined in meters (the distance upwind of the drift trap lip
!   that will not be allowed to accumulate snow).  The model takes
!   this distance and adjusts it to be compatible with the model
!   grid increment (for example, if you give it a 1.0 m distance,
!   and have a 30-m grid increment, you won't get any erosion
!   area).

```

```

!!! (recommended default value: erosion_dist = 0.0)
    erosion_dist = 0.0
!     erosion_dist = 15.0

! Define the Tabler snow accumulation surfaces to include the drift
! traps and some fraction of the initial snow depth and
! precipitation inputs. The "tabler precipitation scale",
! tp_scale, defines the fraction of the initial snow depth and
! precipitation that is added to the Tabler Surfaces. This is
! required so that precipitation accumulating on flat ground, for
! example, is not removed because it is higher than the Tabler
! Surface. Valid numbers are between 0.0 and 1.0.
!!! (recommended default value: tp_scale = 1.0)
!     tp_scale = 0.0
!     tp_scale = 0.25
!     tp_scale = 0.5
!     tp_scale = 0.75
!     tp_scale = 1.0

! For subgrid_flag = 1.0, you have to define the dominant wind
! direction you want Tabler surfaces to be generated. This
! prevents building a drift in one direction, only to have a
! light wind 90 degrees to that taking the drift completely out
! in the next time step. This is just a consequence of how the
! Tabler surfaces are generated and used within the model.
!!! (recommended default value: tabler_dir = 270.0)
    tabler_dir = 270.0

! You can make adjustments to the slopes of the Tabler surfaces
! using the 'slope_adjust' parameter. This factor is multiplied by
! the calculated Tabler equilibrium drift surface slope to get the
! simulated slope. So, for example, if you want the simulated
! slope to be 50% of what Tabler calculated, use a value of 0.5; a
! value of 1.5 would give half again steeper than Tabler's original
! model.
!!! (recommended default value: slope_adjust = 1.0)
    slope_adjust = 1.0

! Define whether you want two-layer accounting for soft (movable)
! and hard (unmovable) layers (flag=0.0=off, flag=1.0=on).
!!! (recommended default value: twolayer_flag = 1.0)
!     twolayer_flag = 0.0
!     twolayer_flag = 1.0

! Define whether you want the upwind boundaries to have zero
! incoming transport flux, or the equilibrium transport flux.
! bc_flag = 0.0 corresponds to a zero flux, and bc_flag = 1.0
! corresponds to the equilibrium flux.
!!! (recommended default value: bc_flag = 0.0)
    bc_flag = 0.0

! Input the height of the wind and rh observations. The model does

```

```

!   not need an air temperature height and is not very sensitive to
!   the relative humidity observation height.
!!! (recommended default value: ht_windobs = user input required)
!!! (recommended default value: ht_rhobs = user input required)
    ht_windobs = 10.0
    ht_rhobs = 3.0

! Snow density. This should be defined to be a typical snow density
! for the SnowTran-3D simulations. The snowpack sub-model will
! actually evolve the snow density over the period of the model
! simulation.
!!! (recommended default value: ro_snow = 300.0)
    ro_snow = 300.0

! Define the initial snow-depth distributions.
!!! (recommended default value: snow_d_init_const = 0.0)
    snow_d_init_const = 0.0

! Define whether the simulation will use a surface topography
! equal to ground topography (topoflag = 0.0), or equal to the
! snow surface (topoflag = 1.0).
!!! (recommended default value: topoflag = 1.0)
    topoflag = 1.0

! Define whether you want the data written out.
! (print_snowtran = 1.0, else 0.0).
!!! (recommended default value: print_snowtran = 1.0)
!!! (recommended default value: snowtran_output_fname =
!!! outputs/snowtran.gdat)
    print_snowtran = 0.0
    snowtran_output_fname = outputs/snowtran.gdat

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!  ENBAL-2D MODEL SETUP
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

! Identify whether the surface energy balance calculation for
! this simulation will include a non-zero conduction term
! (icond_flag = 0 = no conduction, 1 = conduction). Note that
! icond_flag = 1 requires running the multi-layer snowpack model
! with at least 2 snow layers.
!!! (recommended default value: icond_flag = 0)
    icond_flag = 0
!   icond_flag = 1

! Define the albedo for a melting snowcover under the forest
! canopy. This allows the user to control, to some degree, the
! melt rates in the forests. Note that adjusting the gap-fraction
! parameter is also an effective way to do this. The non-melting
! snow albedo is set to 0.8 in the code.

```

```

!!! (recommended default value: albedo_snow_forest = 0.45)
    albedo_snow_forest = 0.45

! Define the albedo for a melting snowcover in non-forested areas.
! This allows the user to control, to some degree, the melt
! rates in non-forested areas. The non-melting snow albedo is
! set to 0.8 in the code.
!!! (recommended default value: albedo_snow_clearing = 0.60)
    albedo_snow_clearing = 0.60

! Define the albedo for a glacier surface (dry and melting).
!!! (recommended default value: albedo_glacier = 0.40)
    albedo_glacier = 0.40

! The code is set up to write out an individual data file for each
! sub-model. Define whether you want the data written out
! (print_enbal = 1.0, else 0.0), and the name of that output
! file.
!!! (recommended default value: print_enbal = 1.0)
!!! (recommended default value: enbal_output_fname =
!!! outputs/enbal.gdat)
    print_enbal = 0.0
    enbal_output_fname = outputs/enbal.gdat

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!! SNOWPACK MODEL SETUP
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

! Define whether static-surface (non-blowing snow) sublimation will
! be included in the model calculations (sfc_sublim_flag = 1.0).
! To turn this off, set sfc_sublim_flag = 0.0. If sfc_sublim_flag
! is turned on, the latent heat flux (Qe) calculated in ENBAL is
! used to add/remove snow from the snowpack.
!!! (recommended default value: sfc_sublim_flag = 1.0)
    sfc_sublim_flag = 1.0

! The code is set up to write out an individual data file for each
! sub-model. Define whether you want the data written out
! (print_snowpack = 1.0, else 0.0), and the name of that output
! file.
!!! (recommended default value: print_snowpack = 1.0)
!!! (recommended default value: snowpack_output_fname =
!!! outputs/snowpack.gdat)
    print_snowpack = 0.0
    snowpack_output_fname = outputs/snowpack.gdat

! Define whether this is going to be a multi-layer snowpack run
! (multilayer_snowpack = 1, else 0). This option does not work
! with the history restart or data assimilation functions.
!!! (recommended default value: multilayer_snowpack = 0)

```

```

!      multilayer_snowpack = 1
!      multilayer_snowpack = 0

! Define the time since last snowfall threshold, in hours. This
! sets how long you have to go without snow precipitation to
! start a new snow layer in the multi-layer snowpack model.
! Periods with zero snowfall at time gaps less than this threshold
! do not create a new snow layer. To force a single layer using
! this parameter, set tsls_threshold >= the number of hours in
! the simulation. Note the equation below can be used to
! calculate this number.
!      tsls_threshold = dt * real(max_iter) / 3600.0
!!! (recommended default value: tsls_threshold = 24.0)
!      tsls_threshold = 1.0
!      tsls_threshold = 24.0
!      tsls_threshold = 12.0
!      tsls_threshold = 144.0
!      tsls_threshold = 8000.0

! Define the maximum number of snow layers you want to allow in
! the multi-layer snowpack model. Note that nz_max in the
! snowmodel.inc file must be at least 1 greater than max_layers.
! So, if you want to run the multi-layer snowpack model with a
! single snow layer, set nz_max=2. If you want to run the
! original single-layer snowpack model, you can set nz_max=1 in
! snowmodel.inc.
!!! (recommended default value: max_layers = 6)
!      max_layers = 12
!      max_layers = 6
!      max_layers = 1

! Define the minimum snow layer thickness (m) for the multi-layer
! snowpack model. 0.001 is an appropriate number. It is also
! okay to set this to be 0.0 which will allow the tracking of
! thin layers in the simulation.
!!! (recommended default value: dz_snow_min = 0.001)
!      dz_snow_min = 0.001
!      dz_snow_min = 0.01
!      dz_snow_min = 0.0

! The code is set up to write out an individual data file for each
! sub-model. Define whether you want the data written out
! (print_multilayer = 1.0, else 0.0), and the name of that output
! file.
!!! (recommended default value: print_multilayer = 1.0)
!!! (recommended default value: multilayer_output_fname =
!!! outputs/multilayer.gdat)
!      print_multilayer = 0.0
!      multilayer_output_fname = outputs/multilayer.gdat

! For multi-year simulations, sometimes it is desirable to zero
! out the snow cover arrays on a certain summer date. For

```

```
!   example, to prevent glaciers from forming.  Input that date
!   here, in the form mmddhh.  Set this to 999999 if you would
!   like to disable this feature.
!!! (recommended default value: izero_snow_date = 999999)
    izero_snow_date = 999999
!   izero_snow_date = 073100
```